DevOps – Lecture 2

Introduction to DevOps - part 2 + Cloud Computing

13 Sept 2022

Chinmaya Dehury

chinmaya.dehury@ut.ee
OUTLINE

DevOps Part 1
- Software Development Life Cycle
- Waterfall model
- Agile model
- DevOps Motivation

DevOps Part 2
- DevOps History & Stats
- What is DevOps and how it works?
- What are Silos and culture?
- DevOps Goals, adoption, values, benefits, etc.
- DevOps Phases (Plan, code, build, test, ...)
- Continuous everything culture in DevOps

Cloud Computing
- Cloud Computing (deployment and service models)
- Virtualizations
- Containerization
- VMs and Containers

Lec-01

This Lecture
A quick recap...
Last Lecture survey results (1/2)

Course Delivery:

For Online delivery:
- 1st: Zoom
- 2nd: Panopto
- 3rd: MS Team
- 4th: BBB
- 5th: Others
Last Lecture survey results (2/2)

DevOps Experience:

- 44% 0 Months
- 16% 1-6 months
- 16% 7-12 months
- 4% 1-2 years
- 0% 2-4 years
- 0% > 4 yrs

Which tools/concepts are you familiar with?

- 16% Git
- 12% JIRA
- 11% Docker
- 14% Kubernetes
- 8% Chef
- 2% Ansible
- 2% Terraform
- 2% Pulumi
- 1% Azure DevOps
- 2% AWS
- 3% Google Cloud
- 7% OpenShift
- 9% Hyperledger
- 9% Mesosphere
- 1% OpenStack
- 4% Other
Software Development Life Cycle (SDLC)

- A lifecycle covers all the stages of software from its inception with requirements definition through to fielding and maintenance [3].
Recap - Agile principles

Simplicity – Maximize the amount of work **not** done

• Most critical thing for Project Manager
  • Having a clear picture of what to build next, not an easy task
  • Maximize, what can be done in next time-window (e.g. in next two weeks)

• Support 80/20 rule [src]
  • aka Pareto principle
  • Approx. 80% of the work can be done in 20% of allocated time
  • Also applicable to testing

• Prioritize the most impactful work
Software Development Life Cycle (SDLC)

Agile Model:

**Disadvantages:**

1. Developer computer is used mainly for testing of those features

2. Test and production environment configuration mismatch

3. Usually not tested in production environment

4. Developers team and operations team are in silos

5. Continuous involvement of all the stakeholders.
Software Development Life Cycle (SDLC)

Agile Model:
Disadvantages:

Developers team
Operations team
Developed product
Product deployment
Production environment
Failed Product
Send product back to developer

Why DevOps?

LTAT.06.015 : Lec-02 : Introduction to DevOps Part II + Cloud Computing
DevOps

Introduction to DevOps - Part 2
Development + Operations

A culture
Development + Operations

Foster the culture
Focus on Outcomes
DevOps History

• Concept of DevOps emerged out of a discussion between Andrew Clay and Patrick Debois in 2008.
Read more at: https://www.appknox.com/blog/history-of-devops

• Devopsdays event*:
  • a worldwide series of technical conferences
  • Topics: software development, IT infrastructure operations, and the intersection between them.
  • Started 2009
  • A series of event in this month (https://devopsdays.org/events)

* this is not an advertisement
DevOps stats-2022

2022 Global DevSecOps Survey by GitLab [direct link]

- 47% respondents follow devops, 34% follow Agile, 26% follow waterfall
- 70% of teams deploy multiple times a day
- 68% of ops teams are “fully” or mostly automated, 56% in 2021
- ~24% of respondents claimed to have full automation
- 51% use AI/ML to check the code
- 60% of developers are releasing code faster than before

Why DevOps?

- Better code quality,
- Developer productivity
- Operational efficiency
DevOps stats

Netflix:
• 1000s of daily production changes
• 10000s of Virtual instances
• 100,000s of customer interactions/sec
• Billions of time-series metrics
• 100s of operation engineers

Amazon:
• Deploys more than 20k releases / day
• 10s of apps

https://www.youtube.com/watch?v=HmM4V33ReCw
https://imgur.com/a/3u8ZKBN
What is DevOps?

• DevOps = Development + Operation
• Evolution of Agile software development Model
• DevOps is not a new technology or a product.
• DevOps is a cultural and professional movement that stresses communication, collaboration and integration between software developers and IT operations professionals.
• Seeks
  • stability
  • Performance
• Speeds up software deliveries to the business.

Image credit: https://aws.amazon.com/devops/what-is-devops/
How DevOps Works?

• Development and Operation teams are no longer siloed (isolated).
• Typically the dev-to-ops ratios was between 6:1 and 8:1, now it is around 25:1 [7]
What is a silo?

Development and Operation teams are no longer siloed (isolated).

What is a silo?

Barriers build up
Different incentives
Different objectives

Disconnection from other people
No shared context
Different management

Bad handoffs
Lack of understanding
Lack of empathy
What is a silo?

How to bring Development and Operation teams to one place?

Communication

Integration

Collaboration
What is a Culture?

- the set of shared attitudes, values, goals, and practices that characterizes an institution or organization
- the integrated pattern of human knowledge, belief, and behavior that depends upon the capacity for learning and transmitting knowledge to succeeding generations

CULTURE IS AN OUTPUT, NOT AN INPUT
DevOps Goals

• Break down silos
• Create cross-functional teams
• Better IT alignment and business responsiveness
• Faster, smaller, more frequent releases
• Improved
  • Time to market
  • Quality of code, products and services
  • Productivity
  • Customer satisfaction
  • Employee satisfaction
• Less waste and fewer defects
• Lower long-term costs
DevOps values - CAMS

• **Culture:**
  • no cultural barriers
  • focus on people not the tools
  • Freedom and shared Responsibility

• **Automation:**
  • Reduce the mundane tasks and automate,
  • continuous delivery
  • IaC
  • lifecycle management with automation of processes, tasks, and decisions

• **Measurements:**
  • measurements in every stage: development, testing, building, integration, monitoring, deployments, etc
  • E.g. Deployment frequency, failure rate, time between failures, time to recovery etc.

• **Sharing:**
  • culture to share knowledge, ideas, tools.
  • Collaboration and communication
  • Feedback
Rule for DevOps Success

• Stay away from “They” culture
• Welcome “We” culture
• Do not play the blame game, take ownership and share resource
  • No punishment
  • Encourage to take risk -> innovation
• Find ways to collaborate – involve others early
• Find ways to automate and make self-service
• Fail fast, continually improve
• Metrics are your friend – always be with measurable outcomes
Where To Start Your DevOps Transformation?

- Improve education, communications, and cross-skilling
- Evaluate new processes to support DevOps
- Re-evaluate and rebuild your service delivery cycle
- Evaluate new tech to support DevOps
- Try it in a small part of the org, or a new project
Worth of reading ...


• How to Deploy DevOps in an Organization
• How to Re-design Systems Toward Continuous Delivery
• How to Assess the Quality of DevOps Practices in Organizations
• Many more.....
DevOps Phases

continous development
continous testing

https://www.edureka.co/blog/devops-lifecycle/

https://medium.com/taptuit/the-eight-phases-of-a-devops-pipeline-fda53lec9bb

https://devopedia.org/devops


https://www.javatpoint.com/devops-lifecycle

https://www.educba.com/devops-lifecycle/
DevOps Phases
DevOps Phases-Plan

Need to have a Vision -> goal for all participants
Idea -> Planning -> Strategy -> Success

What are the thing that should be in the agenda/mind:
• customer, client,
• software requirement,
• enhancement request, bug fix, line-of-business,
• features, idea

Break work down into smaller, manageable chunks for quicker deployments

Addresses
changes and evolutions according to a continuous improvement process and continuous feedback.

Requirements:
Training on tools and metrics in order to have enough visibility of the project’s development.

Tools:
• Jira Software
• Confluence
• Slack

Example:
- Recent incident report on ETAIS
DevOps Phases- Code

Here you mainly
- write code

Use of plugins to
- consistent code-styling
- avoid common security flaws
- Avoid code anti-patterns (e.g. no proper modularize, one solution for all problem, dead code – not sure what it does, unnecessary code/modules/classes, etc).

Several VCS for maintaining the
- codes
- designing infrastructure,
- automating processes,
- defining tests
- implementing security

We will discuss **VCS** (GitLab) in next lecture.
DevOps Phases - Build

Write code ->
1. Commit to the code repo
2. create pull request
3. code review: consistent code-styling, avoid common security flaws, Avoid code anti-patterns.
4. automated tests for possible bugs
5. Merge pull request
DevOps Phases- Test

A test can be:
• Scripted (e.g. test case or automated test) or exploratory
• scenario based testing
• Executed manually or through automation
• Linked to/cover one or more requirements
• Like any other Jira issue type, it can be labeled, prioritized, assigned to components, and commented on
• Use of test environment

Tools:
• Selenium
• TestNG
• JUnit

Types of tests:
• unit tests,
• integration tests,
• functional tests,
• end-to-end tests,
• acceptance tests,
• performance tests,
• and smoke tests.
DevOps Phases- Release

Code successfully passes through all test phases (unit/integration/system tests)

- Build is ready and can be released
- Release can be made automated or manual

From "The Phoenix Project"

Src: https://imgur.com/a/3uBZKBN
DevOps Phases - Deploy

You basically deploy the product on a production environment.
Product is ready for:
- different deployment contexts
- different deployment environments
- different deployment platforms

If the **release process** for deployment to the production environment is also automated, it is referred to as continuous deployment.
DevOps Phases- Operate

The product live in production environment

Operations team relies on automated tools wherever possible for
• configuration management
• scaling
• load balancing
DevOps Phases - Monitor

- Monitor the environments
  - Capturing metrics describing the software’s performance.
  - Collects customers’ feedback on their experiences
  - Analytics on customer behavior
  - Performance, errors, and more
- Monitor the DevOps pipeline itself
Continuous everything culture in DevOps

Continuous Integration
- Develop Code
- Build
- Unit test
- Deploy to Stage
- Functional test
- Deploy to PROD

Continuous Delivery
- Develop Code
- Build
- Unit test
- Deploy to Stage
- Functional test
- Deploy to PROD

Continuous Deployment
- Develop Code
- Build
- Unit test
- Deploy to Stage
- Functional test
- Deploy to PROD

Automatic Trigger
Manual Trigger
Any question so far.....?
Cloud computing
Why are we jumping to Cloud computing now?
Why are we jumping to Cloud computing now?

4% of respondents currently don’t use a cloud provider:
- DevSecOps Survey 2022 by GitLab
Why are we jumping to Cloud computing now?

4% of respondents currently don’t use a cloud provider:  
- DevSecOps Survey 2022 by GitLab

21% of respondents indicated not using the cloud, and instead used a data center or onpremises solution:  
- State of DevOps 2021 by Google
What is Cloud Computing?

• Computing as a utility
  – Utility services e.g. water, electricity, gas, etc.
  – Consumers pay based on their usage

According to NIST Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. (Special Publication 800-145)
What is Cloud Computing?

• Cloud computing is the delivery of shared computing resources (software and/or data) on-demand through the internet.
Cloud deployment model

Cloud deployment includes an emphasize on where the hardware or software is running and who is controlling it.

- Private Cloud
- Community Cloud
- Public Cloud
- Hybrid Cloud

**What do you mean by deployment model?**

- Who is the targeted user?
- Each cloud model is a representative of a specified environment and is distinguished by
  - Size
  - Access
  - Storage and
  - Ownership.

Source: [https://www.netdevgroup.com/online/support/courses/documentation/cloud_and_virtualization_concepts.pdf](https://www.netdevgroup.com/online/support/courses/documentation/cloud_and_virtualization_concepts.pdf)
Cloud deployment models comparison

<table>
<thead>
<tr>
<th>Description</th>
<th>Public Cloud</th>
<th>Private Cloud</th>
<th>Hybrid Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cloud environment</td>
<td>Multi-Tenancy-Shared environment.</td>
<td>Single tenancy-only for single use of an organization.</td>
<td>Both single tenancy and multi–tenancy.</td>
</tr>
<tr>
<td>Data center location</td>
<td>Anywhere</td>
<td>Inside the organization’s network.</td>
<td>Both</td>
</tr>
<tr>
<td>Resource sharing</td>
<td>shared by multiple users in the cloud.</td>
<td>No sharing of resources.</td>
<td>Very secure; integration options add an additional layer of security.</td>
</tr>
<tr>
<td>(Server hardware, network</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>and storage)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

LTAT.06.015 : Lec-02 : Introduction to DevOps Part II + Cloud Computing
Cloud computing is categorized into different service models.

- **Software as a Service (SaaS)**
- **Platform as a Service (PaaS)**
- **Infrastructure as a Service (IaaS)**

**Software as a Service (SaaS)**
- Slack, Office 365

**Platform as a Service (PaaS)**
- AWS Lambda, Google App Engine

**Infrastructure as a Service (IaaS)**
- ETAIS project, Digital Ocean
Virtualization – the underlined technology
Virtualization Platform

• Separation of Virtual Machine from Physical Infrastructure
  • A VM is an isolated runtime environment (guest OS and applications)
  • Multiple virtual systems (VMs) to run on a single physical system

• Benefits of Virtualization Platforms
  • Natural way to deal with the heterogeneity of the infrastructure
  • Allow partitioning and isolating of physical resources
  • Execution of legacy applications

Virtualization Platform (Xen, KVM, VMware…)

Physical Box (Processor, RAM, Memory, Network, Chipset, Storage …)
Distributed Management of VMs

Extending the Benefits of Virtualization to a Physical Cluster

• VM Managers creates a distributed virtualization layer
  • Extend the benefits of VM Monitors from one to multiple resources
  • Decouple the VM (service) from the physical location

Benefits of VM Managers
• Centralized management
• Balance of workload
• Server consolidation
• Dynamic resizing of the infrastructure
• Dynamic cluster partitioning
• Support for heterogeneous workloads
• On-demand provision of VMs
Past, Present & Future of Virtualization

But VMs are:
- Cold start
- Need more storage
  - Less number of virtual machines per PI

PI: Physical Infrastructure
Past, Present & Future of Virtualization

But VMs are
- Cold start
- Need more storage
  - Less number of virtual machines per PI

Containers
The Future of Virtualization
Past, Present & Future of Virtualization

- Lightweight
- Require less memory space
- Fast lunch time
- better resource utilization
- 1000s of containers can be loaded onto a Host

• Containerized apps share Host OS’s kernel to execute work
• Workload in Containers use Host OS kernel

Src: https://www.snia.org/sites/default/files/AnilVasudeva_Containers_the_Future_Virtualization_SDDC.pdf
Past, Present & Future of Virtualization

- Lightweight
- Require less memory space
- Fast lunch time
- better resource utilization
- 1000s of containers can be loaded onto a Host

Will discuss in another Lecture

- Containerized apps share Host OS's kernel to execute work
- Workload in Containers use Host OS kernel

Legacy computing

Virtualization

Containerization

• Containerized apps share Host OS's kernel to execute work
• Workload in Containers use Host OS kernel

Src: https://www.snia.org/sites/default/files/AnilVasudeva_Containers_the_Future_Virtualization_SDDC.pdf
Next

Lec-03: 20 Sept: Version Control System (VCS)
Lab Sessions

**Lab - 01**

- Flask WebApp
  - (Read & Display IoT data from a .csv file)
- VM
- Student's VPC
- UT Private Cloud
Lab Sessions

Lab - 01

Flask WebApp
(Read & Display IoT data from a .csv file)

VM

Student’s VPC

UT Private Cloud

This session Lab - 02

Flask WebApp
(Read & Display IoT data from a .csv file)

Docker Container

VM

Student’s VPC

UT Private Cloud
Lab Sessions

Working with Docker Container Engine
• Installation of docker in your ETAIS Virtual Machine
• Practicing Docker commands
• Building a Dockerfile
• Shipping a Docker image to Docker hub

Lab – 02 : Bonus Task
References

- https://f.ch9.ms/thumbnail/4d7767a7-b218-4aed-8287-a8dfa50a150b.pptx
- https://www.dglobe.com/business/agriculture/4972636-Free-showing-of-Silo-movie-is-March-14
- https://aws.amazon.com/devops/what-is-devops/
- https://www.merriam-webster.com/dictionary/culture
- https://medium.com/ibm-garage/devops-adoption-approach-plan-and-design-be3d1ba67c8
- https://indico.cern.ch/event/32220/contributions/1703574/attachments/623052/857363/An_Introduction_to_Virtualization_and_Cloud_Technologies_to_Support_Grid_Computing_2.0_wc.ppt
- https://www.snia.org/sites/default/files/AnilVasudeva_Containers_the_Future_Virtualization_SDDC.pdf
Any Question?

THANK YOU