ALGEBRA

Applied DevOps

Lecture 01

ABOUT THE COURSE

Teachers

- Teacher:
  - Ivan Radoš, M.A.
  - ivan.rados@predavaci.algebra.hr
  - Contact me via e-mail or Teams
- Consultations: when needed

Goals

- Students will learn:
  - About different workflows and their automation in software development.
  - About continuous integration concepts, tools and workflows.
  - About development and delivery automation concepts.
  - How to use modern tools to implement selected workflows.

Course layout

- The course awards **5 ECTS** points (approx. 30 hours / point)
  - Most of that time will be spent working on projects
- Course layout is the following:

Assessment Name	Assessment Type	Assessment Requirements	Graded or Pass/Fail
Individual Project	Project	Students individually work on the project during the semester. Students are expected to produce a software product and automate its development according to specifications. Allocation of time per student is 90 hours.	Graded
Homework	Practical assessment	The homework aims to encourage students to work independently and to provide their work in a timely manner. In each homework, students will typically have a one or two practical problems. Students are asked to solve each problem and submit the work before the deadline in order to achieve points.	Graded

Hours

Activity:	Total time spent (in hours) across the module:
Lectures	**15**
Independent Study	**105**
Seminars or Tutorials	**30**
TOTAL LEARNING HOURS	**150**

Signature

- In order to obtain a signature, it is necessary to participate in class at the percentage rate, prescribed by the Book of Regulations on studies and studying

Lectures and labs participation	
At least 50 % of physical presence in lectures	At least 60 % of physical presence in labs

- Whoever fails to obtain a signature will have to enroll in the same course the following year, to pay the enrollment and does not have the right to take the exam

Grades

- This course has 100 points available
- Grades:
  - 92,01 – 100,00 points: excellent (5)
  - 75,01 – 92,00 points: very good (4)
  - 58,01 – 75,00 points: good (3)
  - 50,01 – 58,00 points: sufficient (2)

Finishing the course

- This is a project course
- That means that the only way to acquire points is by completing the project

Learning outcomes

Learning Outcome	**Minimal learning outcomes** (after successfully completing the course, student will be able to)	**Wanted learning outcomes** (a successful student should be able to)
I1	Compare different workflows when creating a software solution.	Compare different workflows when creating a larger software solution.
I2	Compare different versioning systems and their use and implement the selected system on a given software solution.	Detect issues in a given versioning system on a given larger software solution.
I3	Compare different systems for monitoring work progress.	Compare different systems for monitoring work progress on a given larger software solution.
I4	Determine the need for continuous integration in the process of creating a software solution.	Determine the need for continuous integration in the process of creating a larger software solution.
I5	Compare available software tools for continuous integration and implement the selected system for continuous integration on the software solution.	Implement the continuous integration chaining on the software solution.

Learning outcomes

Learning Outcome	**Minimal learning outcomes** (after successfully completing the course, student will be able to)	**Wanted learning outcomes** (a successful student should be able to)
I6	Design a system for the delivery of software solutions through a system for continuous integration and apply it in the development of a prototype software solution.	Design a system for the delivery of software solutions and additional artefacts through a system for continuous integration and apply it in the development of a prototype software solution.
I7	Compare available software tools for automation.	Compare available software tools for automation and testing.
I8	Prototype a software solution using systems for automation.	Prototype a software solution using systems for automation and testing.
I9	Suggest appropriate tools for managing and reporting about system parameters and implement selected tools.	Suggest appropriate tools for managing and reporting about system parameters and user interactions and implement selected tools.

Academic standard of conduct

- In written and oral communication it is necessary to follow the rules of business communication appropriate for the academic level
- It is necessary to abide by the strictly defined deadlines for task submissions (homework, seminar papers, projects, etc…)
  - Every task, homework, project etc… , submitted after the defined deadline will not be evaluated nor graded
- Only those students who can confirm their attendance, will be considered as present

Rules of conduct during class – online

- One has to come to class on time
- Disruption of class and inactive participation is not allowed
  - Active participation can be checked by calling out an individual student or by requesting that all (or individual) students click or answer in the system to confirm that they are actively participating in classes
  - Continuous breaking of this rule is sanctioned by reporting students to the Disciplinary Board
- After connecting to Teams, the microphone and camera are turned off and turned on only when asked by the teacher
- In case of question, you click on the question tool and wait for the teacher to call you, after which you turn on the microphone and camera and ask the question

Rules of conduct during class – in classroom

- One has to come to class on time
- Disruption of class and inactive participation is not allowed
  - Continuous breaking of this rule is sanctioned by reporting students to the Disciplinary Board
- Upon entering the classroom, student registers for classes with a card and then sits in an accessible place for work

Rules of conduct during class – in classroom

- Each student must disinfect their hands before accessing the workplace (before entering the space)
- Students are required to wear a mask **properly** all the time. A student who violates that will be removed from class and reported to the Disciplinary Board
- Teachers can remove the mask if they have EU Digital COVID Certificate and more than 2m away from the closest student
- The teacher can come next to the student but they both have to pay attention to the distance and they both must wear a mask
- If all students in a group have EU Digital COVID Certificate, the entire group can remove masks

APPLIED DEVOPS

DevOps definition

- DevOps is a set of practices that brings together software development (Dev) and information technology (IT) (Ops). Its goal is to reduce the length of the development life cycle and to provide continuous delivery of high-quality software. DevOps is a complement to Agile software development; in fact, several aspects of DevOps were derived from the Agile methodology.

DevOps definition

- Academics and practitioners have yet to come up with a universal definition for the term "DevOps," aside from the fact that it is a cross-functional combination of the terms and concepts used to describe "development" and "operations." The key principles of DevOps are shared ownership, workflow automation, and rapid feedback, which are the most frequently observed.

DevOps definition

- A set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production while maintaining high quality, according to Len Bass, Ingo Weber, and Liming Zhu, three computer science researchers from the CSIRO and the Software Engineering Institute, has been proposed as an academic definition of DevOps.

DevOps definition

- However, the term is used in a variety of different contexts. If DevOps is to be truly successful, it must be a combination of specific practices, cultural change, and technological tools.

Agile

The motivations for what has evolved into modern DevOps, as well as several standard DevOps practices such as automated build and test, continuous integration, and continuous delivery, can be traced back to the Agile movement.

Agile development teams employing methods such as Extreme Programming would be unable to satisfy the customer through early and continuous delivery of valuable software.

Agile

The movement to automate operations / infrastructure functions splintered from Agile in the early 2000s and expanded into what is now known as modern DevOps as a result of Scrum becoming the dominant Agile framework in the early 2000s and excluding the engineering practices that were part of many Agile teams.

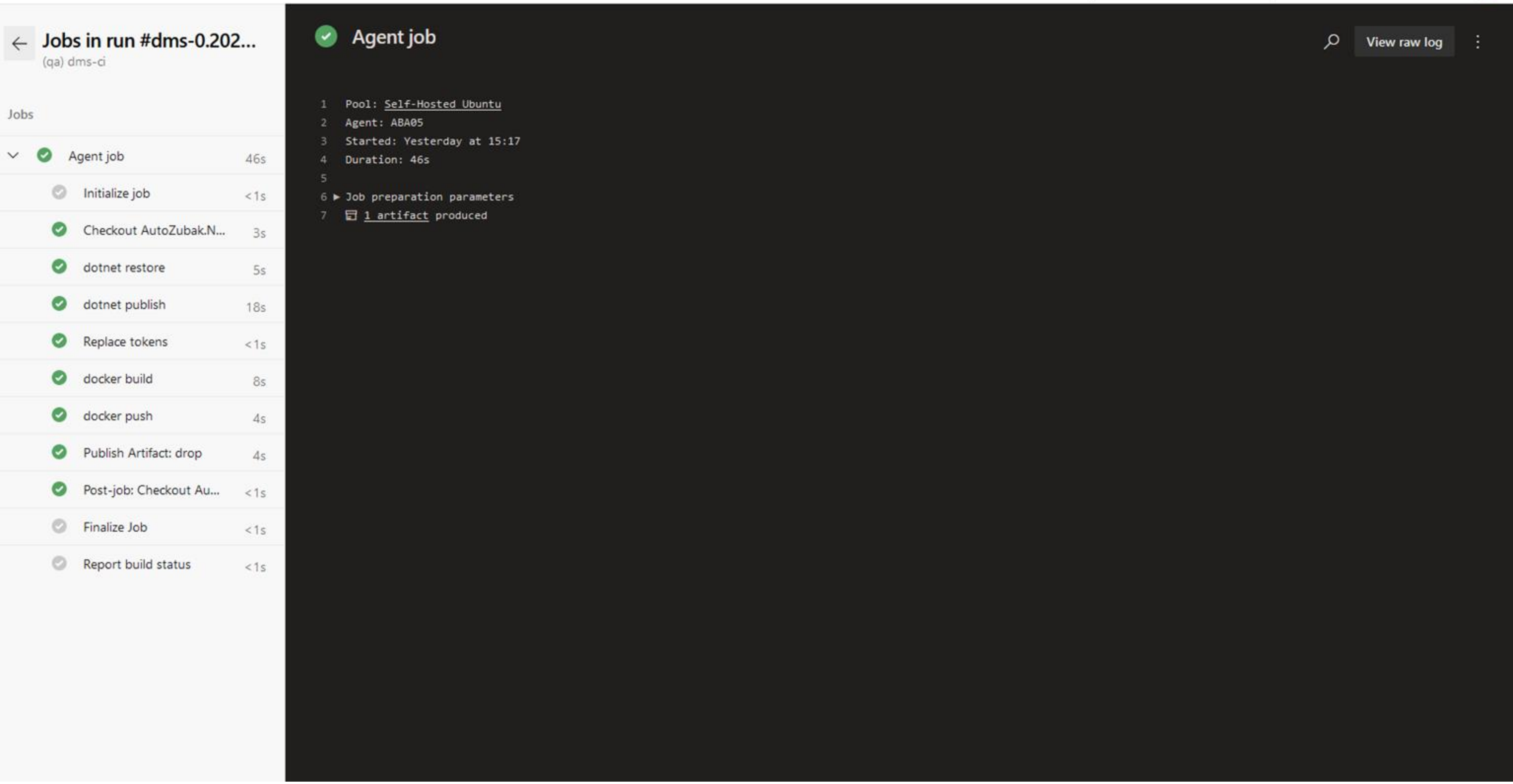
*DevOps is a practice that focuses on the deployment of developed software, regardless of whether it was developed using Agile or another methodology.

CI/CD

In order to achieve DevOps success, automation is a fundamental principle, and continuous integration and delivery (CI/CD) is a key component.

Continued integration (CI) and continuous delivery (CD), or continuous deployment, are two terms used to describe continuous integration and continuous delivery (CI and CD) (CD). When used together, the three processes automate the build, testing, and deployment processes, allowing DevOps teams to ship code changes more quickly and reliably than they could otherwise achieve.

CI/CD



Jobs in run #dms-0.202...

(qa) dms-ci

Jobs

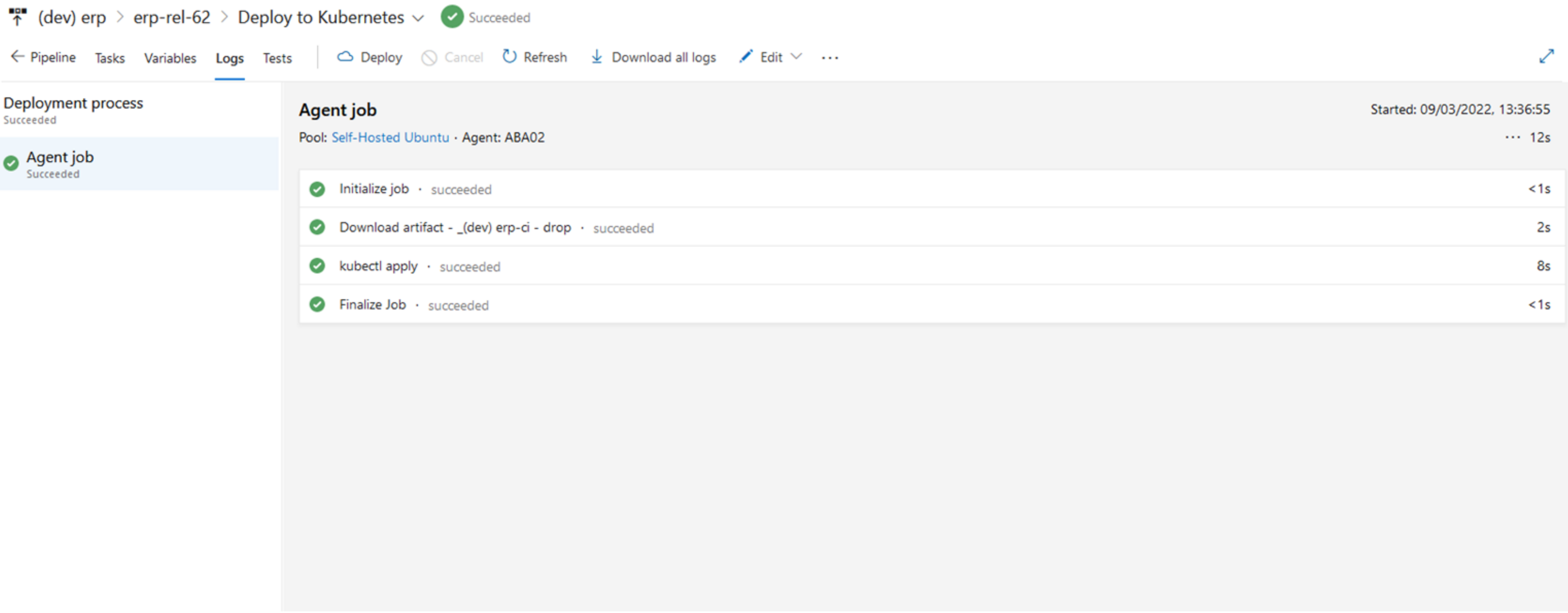
- ✓ Agent job 46s
- Initialize job <1s
- ✓ Checkout AutoZubak.N... 3s
- ✓ dotnet restore 5s
- ✓ dotnet publish 18s
- ✓ Replace tokens <1s
- ✓ docker build 8s
- ✓ docker push 4s
- ✓ Publish Artifact: drop 4s
- ✓ Post-job: Checkout Au... <1s
- Finalize Job <1s
- Report build status <1s

✓ Agent job

```
Pool: Self-Hosted Ubuntu
Agent: ABA05
Started: Yesterday at 15:17
Duration: 46s

▶ Job preparation parameters
1 artifact produced
```

CI/CD



Deployment process
Succeeded

✓ Agent job
Succeeded

Agent job

Started: 09/03/2022, 13:36:55

Pool: Self-Hosted Ubuntu · Agent: ABA02

··· 12s

✓	Initialize job · succeeded	<1s
✓	Download artifact - _(dev) erp-ci - drop · succeeded	2s
✓	kubectl apply · succeeded	8s
✓	Finalize Job · succeeded	<1s

CI/CD

When talking about continuous integration and continuous delivery, the term "CD" is usually used to refer to continuous delivery rather than continuous delivery and delivery. Continuous delivery and other CI/CD processes are primarily concerned with automating software delivery tasks, whereas DevOps is also concerned with organizational change in order to foster excellent collaboration among the many functions involved in the project.

CI/CD

Continuous delivery and other CI/CD processes are primarily concerned with automating software delivery tasks, whereas DevOps is also concerned with organizational change in order to foster excellent collaboration among the many functions involved in the project. Both have a background in agile methods and lean thinking, and both prioritize small and frequent changes that provide focused value to the end customer as a matter of priority. The result is that software is always in a release-ready state throughout its lifecycle, which makes it less expensive and less risky to deliver the software to the customer.

CI/CD

Furthermore, improved collaboration and communication between and within teams contributes to a faster time to market while also reducing risks and reducing expenses.