EAIE

Esports as Accelerator for Social Inclusive Education

LESSON

DEVOPS TOWER BUILDER

AUTHOR

SERGIO ELVIRA PEREZ

CODEARTS SOLUTIONS

DEVOPS TOWER BUILDER

DevOps Tower Builder represents a delivery pipeline as a tower whose levels become stable only when source control, automated tests, container build and deployment stages succeed in sequence. Teams construct and repair the pipeline collaboratively.

Subject and level

Advanced VET learners in software development, cloud infrastructure or systems administration who have basic experience with Git, automated tests and containers.

Connection to curriculum

The exercise supports CI/CD, containerisation, automation, build pipelines, environment configuration and operational feedback. It connects development and infrastructure responsibilities through a shared delivery process.

Didactic and pedagogic considerations

The tower metaphor makes dependencies between pipeline stages visible. Random but controlled failures encourage diagnostic thinking, while team roles prevent a single confident learner from completing the whole pipeline without collaboration.

Time estimate

3 hours

DEVOPS TOWER BUILDER

Group size

15-25 students

Overall introduction to exercise

Development squads construct a five-level pipeline: source, test, build, container and deploy. A level is considered stable only when its automated check passes and the team can explain what would happen if it failed.

The challenge demonstrates that DevOps is a feedback system rather than a single tool. Students experience how automation, documentation and shared ownership reduce risk between writing code and delivering a usable service.

DEVOPS TOWER BUILDER

Exercise as presented to the students

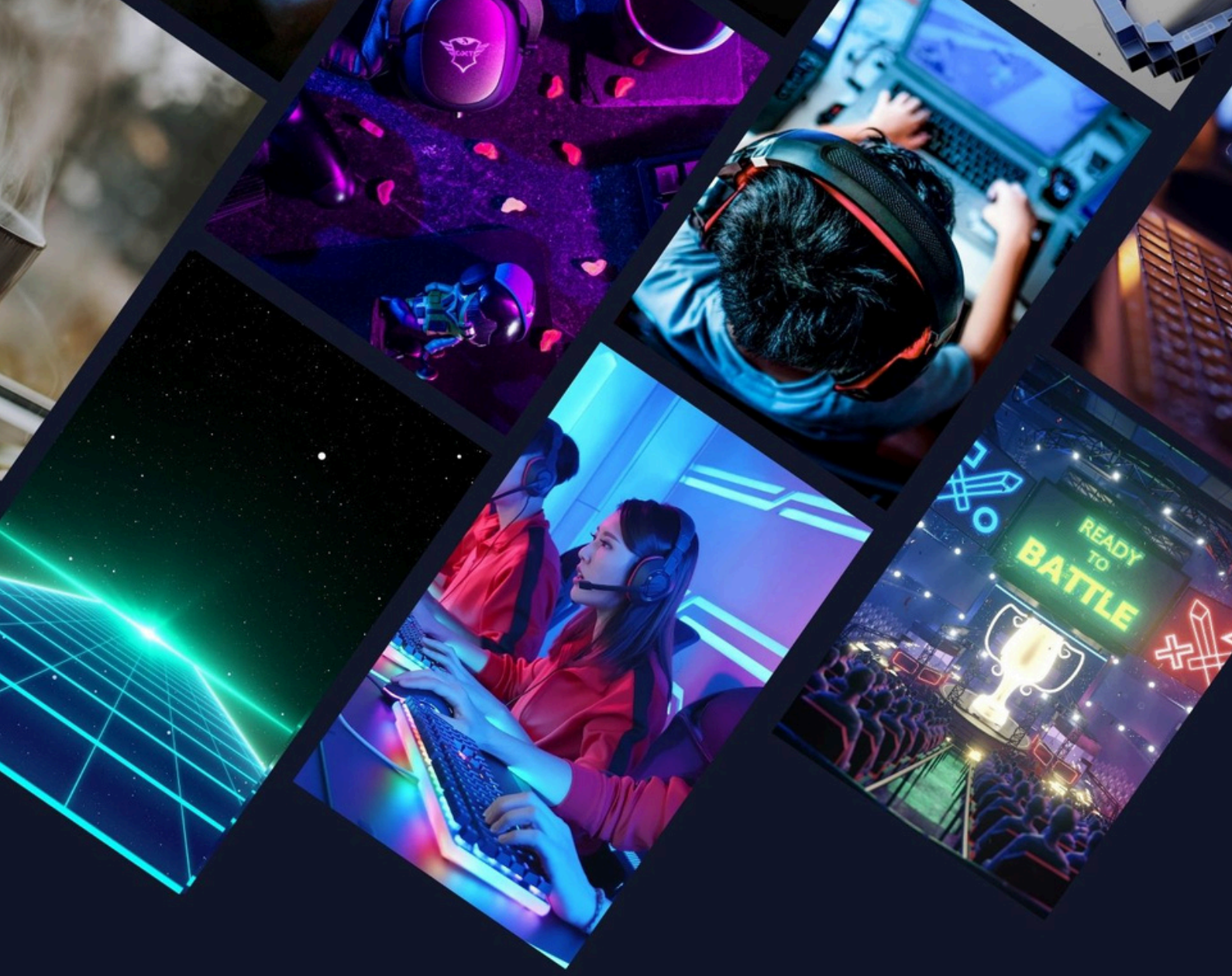
Students formed delivery squads and received a small application, an incomplete Dockerfile and a starter CI configuration. The teacher drew a five-level tower on the board labelled Source, Test, Build, Container and Deploy. Teams could colour in a level only after the corresponding pipeline stage had passed.

The first mission connected repository pushes to an automated unit-test job. Once stable, teams created a container build and verified that the application started from the produced image. A simple local deployment stage then completed the basic tower, with each team recording the purpose and failure signal of every stage.

The teacher introduced controlled production events between rounds: a failing dependency, an incorrect environment variable and a test that intentionally broke after a code change. Teams used job logs and container output to locate the cause rather than rerunning the pipeline repeatedly without investigation.

A bonus mission invited teams to add one useful feedback mechanism such as an artefact, status badge or classroom notification. Solutions were demonstrated to the group and assessed for whether they improved visibility rather than simply adding complexity.

The final reflection mapped the completed tower back to professional CI/CD practice. Students identified where quality gates protected later stages and why developers and operations staff need shared information. The exercise made automation concrete while developing troubleshooting, documentation and collective ownership of delivery.



EASIE
Esports as Accelerator for Social Inclusive Education

E-SPORT AS ACCELERATOR FOR SOCIAL INCLUSIVE EDUCATION

WWW.EASIE.GG

Co-funded by
the European Union

FUNDED BY THE EUROPEAN UNION. VIEWS AND OPINIONS EXPRESSED ARE HOWEVER THOSE OF THE AUTHOR(S) ONLY AND DO NOT NECESSARILY REFLECT THOSE OF THE EUROPEAN UNION OR THE EUROPEAN EDUCATION AND CULTURE EXECUTIVE AGENCY (EACEA). NEITHER THE EUROPEAN UNION NOR EACEA CAN BE HELD RESPONSIBLE FOR THEM.