



Welcome to this session: Introduction to Devops

The session will start shortly...

Questions? Drop them in the chat.
We'll have dedicated moderators
answering questions.



Safeguarding & Welfare

We are committed to all our students and staff feeling safe and happy; we want to make sure there is always someone you can turn to if you are worried about anything.

If you are feeling upset or unsafe, are worried about a friend, student or family member, or you feel like something isn't right, speak to our safeguarding team:



Ian Wyles
Designated Safeguarding
Lead



Simone Botes



Nurhaan Snyman



Rafiq Manan



Ronald Munodawafa



Tevin Pitts

Scan to report a
safeguarding concern



or email the Designated
Safeguarding Lead:
Ian Wyles

safeguarding@hyperiondev.com

Skills Bootcamp Cloud Web Development

- The use of disrespectful language is prohibited in the questions, this is a supportive, learning environment for all - please engage accordingly. **(Fundamental British Values: Mutual Respect and Tolerance)**
- No question is daft or silly - **ask them!**
- There are **Q&A sessions** midway and at the end of the session, should you wish to ask any follow-up questions. Moderators are going to be answering questions as the session progresses as well.
- If you have any questions outside of this lecture, or that are not answered during this lecture, please do submit these for upcoming Academic Sessions. You can submit these questions here: **Questions**

Skills Bootcamp Cloud Web Development

- For all **non-academic questions**, please submit a query:
www.hyperiondev.com/support
- **Report a safeguarding incident:** **www.hyperiondev.com/safeguardreporting**
- We would love your feedback on lectures: [Feedback on Lectures](#)
- If you are hearing impaired, please kindly use your computer's function through Google chrome to enable captions.



What does Continuous Integration help with?

- A. Manually merging code changes
- B. Ensuring new code is automatically tested and integrated frequently
- C. Monitoring application logs in production
- D. Automating cloud infrastructure deployment





Which of the following do you think is the biggest challenge in Software Development?

- A) Writing code quickly
- B) Ensuring that the software runs smoothly in production
- C) Managing a team of developers
- D) Understanding programming languages



Learning Outcomes

- Define DevOps and explain its key principles.
- Describe the differences between traditional software development and DevOps
- Identify and explain common DevOps tools used for CI/CD, containerization, and monitoring.
- Compare and contrast different DevOps practices, such as Continuous Integration vs. Continuous Deployment.

Introduction to DevOps

What is Devops?

- A combination of Development (Dev) and Operations (Ops) to improve collaboration, automation, and efficiency in software delivery.

DevOps

Goals of DevOps:

- Faster and more reliable software releases
- Continuous integration & deployment
- Improved collaboration between development and operations teams

Traditional Software Development vs DevOps

Aspect	Traditional Approach	DevOps Approach
Development & Ops Teams	Separate, isolated	Integrated, collaborative
Release Cycles	Slow & manual	Fast & automated
Deployment Failures	Common, hard to fix	Reduced, quick rollbacks
Infrastructure Setup	Manual	Automated (Infrastructure as Code)

Traditional Vs DevOps

Challenges in Traditional Approaches:

- Slow deployments
- Lack of coordination between teams
- Difficult rollback & debugging processes

How DevOps Solves These Challenges:

- Continuous Integration & Continuous Deployment (CI/CD)
- Infrastructure as Code (IaC)
- Monitoring & Feedback Loops

Core Components of DevOps

1. Continuous Integration (CI)

- Frequent code integration to detect issues early
- Tools: Jenkins, GitHub Actions, GitLab CI/CD

2. Continuous Delivery (CD)

- Automated deployment of code to staging/production
- Tools: Docker, Kubernetes, AWS CodeDeploy

Core Components of DevOps

3. Infrastructure as Code (IaC)

- Automating server and environment setups
- Tools: **Terraform, Ansible, AWS CloudFormation**

4. Monitoring & Logging

- Tracking performance & issues in real-time
- Tools: **Prometheus, Grafana, ELK Stack (Elasticsearch, Logstash, Kibana)**

DevOps Tools and Technologies

Category	Tools
Version Control	Git, GitHub, GitLab
CI/CD	Jenkins, GitHub Actions, CircleCI
Containerization	Docker, Kubernetes
Infrastructure as Code	Terraform, Ansible
Monitoring & Logging	Prometheus, Grafana, ELK Stack

Core Components of DevOps

- **Plan** – Define project goals & requirements
- **Develop** – Code & commit changes (Git, GitHub)
- **Build** – Automate builds using CI/CD pipelines
- **Test** – Run automated tests
- **Release** – Deploy to staging or production
- **Monitor** – Use monitoring tools to track performance

Conclusion and Q&A

Key Takeaways:

- DevOps improves collaboration, automation, and efficiency
- CI/CD pipelines help automate software delivery
- Infrastructure as Code simplifies environment management

Discussion Questions

How does DevOps differ from traditional development methods?

Discussion Questions

- **What challenges do teams face when adopting DevOps?**

Questions and Answers



Thank you for attending



CoGrammar



Department
for Education