

DEVOPS ENGINEER

Details of standard

Occupation summary

This occupation is found in a wide and diverse range of public and private sector organisations, from tech start ups through government departments to multinationals. Essentially any organisation of any size that builds and/or operates modern IT services.

The broad purpose of the occupation is to enable organisations to get valuable working software out in front of active users, both external and internal, frequently and safely, reducing time to market, delivering increased value - both with respect to the end user and the business - and improving the quality of digital services. At its simplest, DevOps is a philosophy and way of working that brings together two historically disparate parts of the IT organisation, namely those who develop the software and those who are then required to support it in the live environment. The DevOps Engineer encapsulates both disciplines, requiring the individual to understand and appreciate how their code functions when being used in the real world and troubleshoot any issues that may arise, while taking a cloud-infrastructure focused perspective. This means taking responsibility for all aspects of the development and operations process - the design, build, test, implement, release and continual iteration of products. Utilizing the advantages of Cloud computing to enable infrastructure to be defined in code moves the operations side away from traditional system administrator roles which are focused on troubleshooting traditional infrastructure-as-hardware. The convergence of these two topics drives DevOps culture and ways of working and creates the need for the new role of DevOps Engineer that works within the delivery team. The DevOps Engineer applies all the DevOps culture and software engineering disciplines to codified infrastructure.

In their daily work, an employee in this occupation interacts with other members of agile development teams, other areas within the organisation's IT department and business areas, as well as 3rd-party suppliers. This is an office based or remote working role, with co-location preferable.

An employee in this occupation will be responsible for working collaboratively with a minimum of direct supervision within broad but generally well defined parameters, requiring the application of knowledge and understanding, skills and methods in a broad range of complex or technical work activities, performed in a variety of contexts. They will address problems which are non-routine while normally fairly well defined, taking responsibility for courses of action, including, where relevant, responsibility for the work of others and allocation of resources. They will typically pair with other technical roles, or work alone at times, and provide input to the planning of work and advise on design.

Typical job titles include:

DevOps Engineer, Infrastructure Engineer, Platform Engineer, Reliability Engineer, Site Reliability Engineer, Build and Release Engineer, Automation Engineer, Full Stack Developer, Deployment Engineer

Entry requirements

Individual employers will set the selection criteria, but this is likely to include A levels; a level 3 apprenticeship or other relevant qualification; relevant experience and/or an aptitude test.

Occupation duties

Duty

Duty 1 Script and code in at least one general purpose language and at least one domain-specific language to orchestrate infrastructure, follow test driven development and ensure appropriate test coverage.

Duty 2 Initiate and facilitate knowledge sharing and technical collaboration with teams and individuals, with a focus on supporting development of team members.

Duty 3 Engage in productive pair/mob programming to underpin the practice of peer review.

Duty 4 Work as part of an agile team, and explore new ways of working, rapidly responding to changing user needs and with a relentless focus on the user experience. Understand the importance of continual improvement within a blameless culture.

Duty 5 Build and operate a Continuous Integration (CI) capability, employing version control of source code and related artefacts.

Duty 6 Implement and improve release automation & orchestration, often using Application Programming Interfaces (API), as part of a continuous delivery and continuous deployment pipeline, ensuring that team(s) are able to deploy new code rapidly and safely.

Duty 7 Provision cloud infrastructure using APIs, continually improve infrastructure-as-code, considering use of industry leading technologies as they become available (e.g. Serverless, Containers).

Duty 8 Evolve and define architecture, utilising the knowledge and experience of the team to design in an optimal user experience, scalability, security, high availability and optimal performance.

KSBs

K4 K5 K6 K7 K8 K11 K12 K13 K14 K20

S9 S11 S13 S14 S16 S17 S18 S20 S22

B2 B3

K4 K9 K10 K18 K19 K23

S1 S2 S4 S8 S16 S21

B1 B2 B4

K4 K6 K7 K9 K20

S1 S8 S13 S14 S17 S18

B1 B3

K4 K6 K9 K10 K18 K19 K22 K23

S1 S2 S3 S4 S8 S11 S21

B1 B3 B4

K1 K2 K7 K14 K15 K24

S4 S9 S12 S15 S16 S18 S20

B3

K1 K2 K7 K15 K17 K24

S4 S5 S9 S12 S15 S16 S18 S20

B3

K1 K2 K3 K7 K8 K11 K12 K13 K14 K15
K17 K21 K24

S4 S5 S6 S9 S12 S15 S16 S18 S21 S22

B3

K1 K3 K4 K6 K7 K8 K9 K10 K11 K12
K15 K21 K24

S1 S2 S4 S8 S9 S10 S11 S15 S16 S17
S18 S21 S22

B1 B3

Duty 9 Apply leading security practices throughout the Software Development Lifecycle (SDLC).

K3 K4 K5 K6 K7 K8 K9 K10 K11 K12
K16

S3 S4 S9 S10 S11 S14 S16 S17 S18

B1 B3

Duty 10 Implement a good coverage of monitoring (metrics, logs), ensuring that alerts are visible, tuneable and actionable.

K4 K6 K7 K10 K11 K13 K24

S3 S4 S6 S7 S9 S11 S16 S17 S18 S19

B3

Duty 11 Keep up with cutting edge by committing to continual training and development - utilise web resources for self-learning; horizon scanning; active membership of professional bodies such as Meetup Groups; subscribe to relevant publications.

K4 K9 K10 K22 K23 K24 K25

S4 S16

B1 B2

Duty 12 Look to automate any manual tasks that are repeated, often using APIs.

K1 K4 K6 K7 K8 K11 K13 K14 K15 K17

S4 S5 S7 S12 S15 S17 S18

B3

Duty 13 Accept ownership of changes; embody the DevOps culture of 'you build it, you run it', with a relentless focus on the user experience.

K4 K6 K9 K10 K22 K23 K24

S1 S2 S4 S6 S7 S8 S11 S17 S18

B1 B3 B4

KSBs

Knowledge

K1: Continuous Integration - the benefits of frequent merging of code, the creation of build artefacts and ensuring all tests pass, with automation throughout - including common tooling.

K2: The principles of distributed Source Control, including how to exploit the features of the tool, such as branching.

K3: How to use data ethically and the implications for wider society, with respect to the use of data, automation and artificial intelligence within the context of relevant data protection policy and legislation.

K4: The business value of DevOps in terms of Time, Cost, Quality, with an emphasis on building in internal Quality throughout the lifetime of the product.

K5: A range of modern security tools and techniques - e.g. threat modelling, vulnerability scanning and dependency checking, with a general awareness of penetration testing - in order to deal with threats and attack vectors within code and across the cyber domain.

K6: A range of problem solving techniques appropriate to the task at hand, such as affinity mapping, impact maps, plan-do-check-act/Deming.

K7: General purpose programming and infrastructure-as-code.

K8: Immutable infrastructure and how it enables continuous refreshing of software, namely the updating of the operating system, container and security patching.

K9: Different organisational cultures, the development frameworks utilised and how they can both complement each other and introduce constraints on delivery.

K10: How the user experience sits at the heart of modern development practices in terms of strategies to understand diverse user needs, accessibility and how to drive adoption.

K11: Monitoring and alerting technologies and an awareness of the insights that can be derived from the infrastructure and applications - collecting logs and metrics, configuring alerting thresholds, firing alerts and visualising data.

K12: The persistence/data layer, including which database/storage technologies are appropriate to each platform type and application when considering non-functional and functional needs; e.g. monolith, microservice, read heavy, write heavy, recovery plans.

K13: Automation techniques, such as scripting and use of APIs.

K14: Test Driven Development and the Test Pyramid. How the practice is underpinned by unit testing, the importance of automation, appropriate use of test doubles and mocking strategies, reducing a reliance on end-to-end testing.

K15: The principles and application of Continuous Integration, Continuous Delivery and Continuous Deployment, including the differences between them.

K16: How best to secure data; e.g. encryption in transit, encryption at rest and access control lists (ACL).

K17: What an API is, how to find them and interpret the accompanying documentation.

K18: Roles within a multidisciplinary team and the interfaces with other areas of an organisation.

K19: Different methods of communication and choosing the appropriate one - e.g. face-to-face (synchronous, high bandwidth), instant messaging, email (asynchronous, low bandwidth), visualisations vs. words.

K20: Pair/mob programming techniques and when to use each technique.

K21: Architecture principles, common patterns and common strategies for translating user needs into both cloud infrastructure and application code.

K22: How their occupation fits into the wider digital landscape and any current or future regulatory requirements.

K23: The importance of continual improvement within a blameless culture.

K24: The difference between Software-as-a-Service (SaaS) v bespoke v enterprise tooling and how to make an informed choice that suits each use case.

K25: Maintain an awareness of cloud certification requirements.

Skills

S1: Communicate credibly with technical and non-technical people at all levels, using a range of methods; e.g. 'Show and Tell' and 'Demonstrations'.

S2: Work within different organisational cultures with both internal and external parties

S3: Translate user needs into deliverable tasks, writing clear, concise and unambiguous user stories that the whole team can understand.

S4: Initiate and facilitate knowledge sharing and technical collaboration

S5: Deploy immutable infrastructure

S6: Install, manage and troubleshoot monitoring tools

S7: Navigate and troubleshoot stateful distributed systems, in order to locate issues across the end-to-end service.

S8: Work in agile, multi-disciplinary delivery teams, taking a flexible, collaborative and pragmatic approach to delivering tasks.

S9: Application of a range of cloud security tools and techniques - e.g. threat modelling, vulnerability scanning, dependency checking, reducing attack surface area - incorporating these tools and techniques into the automated pipeline wherever possible.

S10: Assess identified and potential security threats and take appropriate action based on likelihood v impact.

S11: Employ a systematic approach to solving problems, using logic and hypotheses / experimentation to identify the source of issues.

S12: Automate tasks where it introduces improvements to the efficiency of business processes and reduces waste, considering the effort and cost of automation.

S13: Engage in productive pair/mob programming.

S14: Write tests and follow Test Driven Development discipline in various different contexts.

S15: Release automation and orchestration as part of a Continuous Integration workflow and Continuous Delivery pipeline, automating the delivery of code from source control to the end users.

S16: Invest in continuous learning, both your own development and others, ensuring learning activities dovetail with changing job requirements. Keep up with cutting edge.

S17: Code in a general purpose programming language.

S18: Specify cloud infrastructure in an infrastructure-as-code domain-specific language.

S19: Interpret logs and metrics data within the appropriate context to identify issues and make informed decisions.

S20: Writing code in such a way that makes merging easier and facilitates branching by abstraction - i.e. feature toggling.

S21: Application of lightweight modelling techniques, such as whiteboarding, in order to gain consensus as a team on evolving architecture.

S22: Incremental refactoring by applying small behaviour-preserving code changes to evolve the architecture.

Behaviours

B1: Exhibits enthusiasm, openness and an aptitude for working as part of a collaborative community; e.g. sharing best practice, pairing with team members, learning from others and engaging in peer review practices.

B2: Invests time and effort in their own development, recognising that technology evolves at a rapid rate.

B3: Displays a commitment to the mantra 'You build it, you run it', taking ownership of deployed code and being accountable for its continual improvement, learning from experience and taking collective responsibility when things fail.

B4: Is inclusive, professional and maintains a blameless culture.

Qualifications

English & Maths

Apprentices without level 2 English and maths will need to achieve this level prior to taking the End-Point Assessment. For those with an education, health and care plan or a legacy statement, the apprenticeship's English and maths minimum requirement is Entry Level 3. A British Sign Language (BSL) qualification is an alternative to the English qualification for those whose primary language is BSL.

Additional details

Occupational Level:

4

Duration (months):

24

Review

This apprenticeship standard will be reviewed after three years

Find an apprenticeship

Version log

VERSION	DATE UPDATED	CHANGE	PREVIOUS VERSION
1	20/03/2020	Minor amendment. Weighting has been amended on page 11 to reflect the overall grade table.	Previous version
1	12/03/2020	Assessment plan first published. Approved for delivery	Not available
1	12/03/2020	Updated structure	Previous version
1	07/02/2020	Funding band first published	Not available
1	20/09/2019	Standard first published	Not available
1	03/10/2018	Initial creation	Not available