

Network Admin to Site Reliability Engineer

SKILLSOFT ASPIRE JOURNEY

skillsoft ▶▶

Głównym wyzwaniem przed którym stają dziś organizacje na całym świecie jest konieczność ciągłego podnoszenia umiejętności i poziomu wiedzy w ślad za gwałtownym rozwojem nowych technologii i zmian na globalnym rynku.

Stały rozwój i podnoszenie kwalifikacji w IT od dawna jest już rzeczą oczywistą, a możliwość zapewnienia wsparcia specjalistom chcącym stale się rozwijać jest jedną z głównych kart przetargowych w walce o pracownika.

Na rynku liczą się dziś ludzie, którzy posiadają konkretne kompetencje i zestaw umiejętności pozwalający im wykonywać zadania efektywnie, a nie Ci z najdłuższym stażem pracy.

Dziś, bardziej niż kiedykolwiek w cenie jest umiejętność budowania ścieżki kariery dla profesjonalistów IT, którzy wciąż chcą się liczyć na rynku pracy.

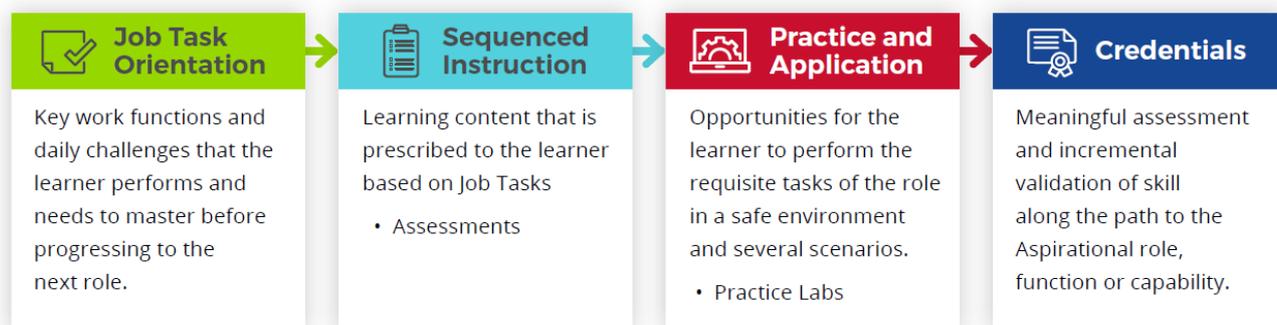
Skillsoft Aspire Journey stanowi odpowiedź na pytanie, jakie szkolenia muszą ukończyć, aby być przygotowanym do swojej wymarzonej pracy. Spośród kilkuset kanałów tematycznych dostępnych na naszej platformie szkoleniowej nasi specjaliści wybrali te, które naszym zdaniem najlepiej wyposażą uczących się w narzędzia potrzebne do realizacji zadań w nowej roli.

Skillsoft Aspire Journey to zestawy szkoleń i ćwiczeń w języku angielskim, które metodycznie, krok po kroku pozwalają specjalistom przejść od poziomu podstawowego do zaawansowanego.

Każda ścieżka zawiera szkolenia, laboratoria wirtualne, video i książki, które pomogą uczącym się osiągnąć pożądane kompetencje poświadczony certyfikatem.

Aspire Journey Model

Cała ścieżka opiera się na 4-elementowym cyklu powtarzanym na kolejnych etapach nauki.



1. Określenie kluczowych funkcji i wyzwań, z którymi musi poradzić sobie uczący się w chwili obecnej, jak i tymi, z którymi przyjdzie mu się zmierzyć w nowej pracy.
2. Przejście zaprojektowanych ścieżek w proponowanej kolejności, wykonanie ćwiczeń i zaliczenie testów.
3. Przećwiczenie nowych umiejętności w kontrolowanym środowisku w oparciu o gotowe scenariusze działań. Laboratoria wirtualne Skillsoft
4. Certyfikat – zaliczenie testu końcowego na poziomie co najmniej 70% i uzyskanie certyfikatu potwierdzającego ukończenie danego etapu nauki.

Aspire Journeys 

Network Admin to Site Reliability Engineer

Site Reliability Engineering is a combination of software engineering and IT Operations to help create scalable and reliable software systems. In this Skillsoft Aspire journey, you will explore the skills required to go from a Network Admin, a DevOps E...

[View More](#) ▾

 49 courses | 49h 29m 14s  4 labs | 32h

 Earn a Badge

Analizując trendy opisujące zachowanie użytkowników na naszych platformach szkoleniowych i współpracując ściśle z naszymi klientami na całym świecie Skillsoft wyselekcjonował najlepsze materiały szkoleniowe i ułożył je w ustrukturalizowaną ścieżkę rozwoju. Całość zawiera ponad 49 godzin szkoleniowych.

Tracks

	<p>Track 1: Network Admin</p> <p>In this track of the Network Admin to Site Reliability Engineer Skillsoft Aspire journey, the focus will be on OS deployment, backup and recovery, monitoring distributed systems, and SRE scenario...</p> <p>View More</p> <p>Explore  12 courses 10h 38m 53s  1 lab 8h</p>		
	<p>Track 2: DevOps Engineer</p> <p>In this track of the Network Admin to Site Reliability Engineer Skillsoft Aspire journey, the focus will be on build & release engineering best practices, automation and simplicity best practices for ...</p> <p>View More</p> <p>Explore  11 courses 11h 38m 21s  1 lab 8h</p>		
	<p>Track 3: Chaos Engineer</p> <p>In this track of the Network Admin to Site Reliability Engineer Skillsoft Aspire journey, the focus will be on troubleshooting, emergency response and incident handling, testing for reliability, load ...</p> <p>View More</p> <p>Explore  17 courses 18h 20m 12s  1 lab 8h</p>		
	<p>Track 4: Site Reliability Engineer</p> <p>In this track of the Network Admin to Site Reliability Engineer Skillsoft Aspire journey, the focus will be on scaling the SRE team, operational loads, communication and collaboration and managing sof...</p> <p>View More</p> <p>Explore  9 courses 8h 51m 46s  1 lab 8h</p>		

PREREQUISITES

In order to fully profit from the potential of this Aspire Journey, we recommend the following prerequisite skills:

- Familiar with Site Reliability Engineering
- Familiar with Networking
- Familiar with DevOps

Track 1: Network Admin

In this track of the Network Admin to Site Reliability Engineer Skillssoft Aspire journey, the focus will be on OS deployment, backup and recovery, monitoring distributed systems, and SRE scenario planning.

▶ 12 courses | 10h 38m 53s ◀ 1 lab | 8h



Site Reliability: Engineering

Objectives

- provide an overview of Site Reliability Engineering
- recognize the nine principles of Site Reliability Engineering
- list the core tenets of SRE
- differentiate between SRE and DevOps
- provide an overview Service Level Indicators
- provide an overview of Service Level Objectives
- provide an overview of Service Level Agreements
- recognize how to embrace and manage risk in an environment
- recognize how to measure service risk using metrics such as time-based availability and aggregate availability
- identify the risk tolerance of infrastructure services
- provide an overview of error budgets



Site Reliability: Tools & Automation

Objectives

- provide an overview of planning tools such as JIRA and Pivotal Tracker
- differentiate between tools used for creation such as GitHub and Subversion
- describe common tools used for packaging and releasing services and releases
- differentiate between different tools used to automate functions
- provide an overview of tools used to monitor applications and infrastructure
- describe the value of automation including consistency, platform, repairs, and time savings
- provide an overview of uses cases for automation
- describe the path that the evolution of automation follows
- describe how automation processes can vary
- provide an overview of common pitfalls associated with troubleshooting systems
- provide an overview of the primary goals of a post-mortem philosophy
- determine which factors are the root cause of a problem



**OS Deployment
Strategies:
Upgrading &
Maintaining Systems**

Objectives

- recognize the importance of upgrading and patching systems
- list Windows 10 upgrade and migration considerations
- provide an overview of Windows 10 edition upgrade
- provide an overview of Windows 10 upgrade paths
- differentiate between modern deployment options such as Windows Autopilot and In-place upgrades
- differentiate between dynamic deployment options such as subscription activation, AAD/MDM, and provisioning packages
- describe traditional deployment options such as bare metal, refresh, and replace
- migrate files and settings using the User State Migration Tool
- provide an overview of the Windows Assessment and Deployment Kit
- provide an overview of the Microsoft Deployment Toolkit
- provide an overview of Windows to Go
- recognize how Windows Update works
- recognize Windows 10 servicing and support features
- provide an overview of Windows Servicing Channels



**OS Deployment
Strategies: Deploying
Modern Systems**

Objectives

- differentiate between on-premise and cloud deployments
- provide an overview of public clouds
- provide an overview of private cloud
- provide an overview of hybrid cloud
- provide an overview of community cloud
- provide an overview of the IaaS service model
- provide an overview of the PaaS service model
- provide an overview of the SaaS service model
- provide an overview of Identity as a Service
- provide an overview of Network as a Service



**OS Deployment
Strategies:
Maintaining &
Managing Modern
Systems**

Objectives

- provide an overview of Windows Group Policy
- describe how to deploy Windows 10 updates using Windows Server Update Services
- provide an overview of Windows Update for Business
- differentiate between the two methods of peer-to-peer content distribution in Windows 10 - Delivery Optimization and BranchCache
- provide an overview of Express Update Delivery in Windows 10
- describe how to monitor Windows Updates with Update Compliance
- recognize how to deploy updates for Windows 10 Mobile Editions
- provide an overview of key concepts relating to Windows as a Service
- provide an overview of Desktop Analytics
- provide an overview of dynamic provisioning



Backup & Recovery: Business Continuity & Disaster Recovery

Objectives

- describe system resilience and fault tolerance
- recognize what data to include in a backup, such as full operating system, personal files, and company data
- identify and classify assets such as PII, PHI, and proprietary data
- provide an overview of data classifications such as Top Secret, Secret, Confidential, and Unclassified
- recognize the financial impact of a large data breach or cyber incident
- provide an overview of performing a business impact assessment
- recognize different types of risks to consider
- describe elements of continuity planning such as strategy and processes
- define appropriate service levels
- provide an overview of data protection solutions that can be used to protect against security issues and reduce the risk of compromised data
- provide an overview of common roles and responsibilities of an administrator when it comes to data backups and recovery
- provide an overview of Recovery Time Objective and Recovery Point Objective and how they can relate to a timely recovery
- describe elements to include when preparing a disaster recovery training plan



Backup & Recovery: Enterprise Backup Strategies

Objectives

- differentiate between onsite and offsite backup solutions and describe the benefits of each
- provide an overview of the different types of backups
- describe considerations when backing up BYOD devices
- recognize the cultural impact of moving to the cloud
- describe an example backup strategy such as 3-2-1
- provide an overview of AWS backup and restore services
- provide an overview of Microsoft Azure for backups
- provide an overview of cloud storage solutions offered by Google
- describe the configuration and benefits of a hybrid cloud solution



Backup & Recovery: Windows Client Backup and Recovery Tools

Objectives

- enable and configure the File History backup feature in Windows 10
- use the Backup and Restore feature and describe how it differs from the File History feature
- create a System Image Backup in Windows 10
- describe and use the OneDrive solution in Windows 10
- create a system restore point in Windows 10
- use the Advanced Startup Options to fix a Windows 10 PC
- enable volume shadow copies in Windows 10
- create a recovery drive to enable access to the advanced startup options
- use the Reset this PC feature
- backup and restore favorites from Microsoft Edge
- revert a system to the previous working version of Windows
- remove a Windows update to restore working order of a system



Describing Distributed Systems

Objectives

- provide an overview of distributed systems
- provide an overview of the four basic architecture models - client/server, three-tier, n-tier, and peer-to-peer
- describe the benefits of monitoring including tracking long term trends, testing performance, alerting, business analytics, and improved debugging
- describe the importance of monitoring such as analyzing long-term trends, alerting, and building dashboards
- differentiate between whitebox and blackbox monitoring
- provide an overview of monitoring dashboards including what they are, why we use them, and common elements of a dashboard
- differentiate between events, alerts, and incidents
- describe the importance of conducting ad hoc retrospective analysis
- recognize best practices when it comes to general system monitoring
- recognize considerations for a successful monitoring strategy for establishing a reliable network
- describe the importance of setting reasonable expectations for monitoring



Monitoring Distributed Systems

Objectives

- provide an overview of nodes and machines
- provide an overview of root cause
- differentiate between symptoms and causes
- provide an overview of the 'four golden signals' of monitoring
- describe the importance of focusing on monitoring traffic and how it applies to the four golden signals
- describe the importance of focusing on the 'errors' metric and how it applies to the four golden signals
- provide an overview of saturation and how it applies to the four golden signals
- recognize strategies for effective monitoring and addressing mean values
- choose an appropriate resolution for measurements
- list guidelines to keep in mind when designing a monitoring system
- determine the appropriate questions to ask when creating rules for your monitoring solution
- recognize the importance of how decisions about monitoring are made with long terms goals



Site Reliability
Engineering:
Scenario Planning

Objectives

- define scenario planning and identify why it should be part of your strategic plan
- describe how to use scenario planning and how to create scenarios
- recognize considerations when scenario planning for a disaster
- identify potential scenarios to test and prepare for, such as the loss of technical infrastructure or environmental issues
- list common data-related disaster recovery scenarios to plan for
- list common applications-related disaster recovery scenarios to plan for
- provide an overview of disaster recovery testing events and how they can help identify vulnerabilities in critical systems
- list what to test when designing tests for DiRT
- recognize how to minimize the potential damage of disruptive DiRT tests
- provide an overview of the DiRT technical team and the coordination team
- list common components of a DiRT test plan and how creating a template is useful for future test plan proposals
- outline the functions of a Customer Reliability Engineering team and their role in scenario planning
- outline the production incident lifecycle and how to lay the foundations to shrink production incidents
- provide an overview of unmanaged responses
- describe how to rectify untrained responses
- recognize hands-on activities used to train response teams
- describe how DiRT exercises should also test how people organize themselves and interact with each other
- provide an overview of the "Wheel of Misfortune" role-playing scenario
- provide an overview of the Dungeon/Scenario Master and their role in running a test scenario



Final Exam: Network Admin

Objectives

- create a System Image Backup in Windows 10
- create a system restore point in Windows 10
- define appropriate service levels
- define scenario planning and identify why it should be part of your strategic plan
- describe and use the OneDrive solution in Windows 10
- describe an example backup strategy such as 3-2-1
- describe appropriate service levels
- describe common tools used for packaging and releasing services and releases
- describe considerations when backing up BYOD devices
- describe how to deploy Windows 10 updates using Windows Server Update Services
- describe how to monitor Windows Updates with Update Compliance
- describe how to rectify untrained responses
- describe how to use scenario planning and how to create scenarios
- describe system resilience and fault tolerance
- describe the importance of focusing on the 'errors' metric and how it applies to the four golden signals
- describe traditional deployment options such as bare metal, refresh, and replace
- differentiate between different tools used to automate functions
- differentiate between events, alerts, and incidents
- differentiate between on-premise and cloud deployments
- differentiate between onsite and offsite backup solutions and describe the benefits of each
- differentiate between SRE and DevOps
- differentiate between symptoms and causes
- differentiate between the two methods of peer-to-peer content distribution in Windows 10 - Delivery Optimization and BranchCache
- differentiate between tools used for creation such as GitHub and Subversion
- differentiate between whitebox and BlackBox monitoring
- enable and configure the File History backup feature in Windows 10
- explain system resilience and fault tolerance
- list the core tenets of SRE
- list what to test when designing tests for DiRT
- list Windows 10 upgrade and migration considerations
- provide an overview of data classifications such as Top Secret, Secret, Confidential, and Unclassified
- provide an overview of disaster recovery testing events and how they can help identify vulnerabilities in critical systems
- provide an overview of Express Update Delivery in Windows 10
- provide an overview of hybrid cloud
- provide an overview of monitoring dashboards including what they are, why we use them, and common elements of a dashboard
- provide an overview of nodes and machines
- provide an overview of planning tools such as JIRA and Pivotal Tracker
- provide an overview of public clouds
- provide an overview of Service Level Agreements
- provide an overview of Site Reliability Engineering
- provide an overview of the different types of backups

- provide an overview of the four basic architecture models - client/server, three-tier, n-tier, and peer-to-peer
- provide an overview of the 'four golden signals' of monitoring
- provide an overview of the IaaS service model
- provide an overview of the PaaS service model
- provide an overview of the root cause
- provide an overview of tools used to monitor applications and infrastructure
- provide an overview of unmanaged responses
- provide an overview of Windows 10 edition upgrade
- provide an overview of Windows 10 upgrade paths
- provide an overview of Windows Group Policy
- recognize best practices when it comes to general system monitoring
- recognize considerations when scenario planning for a disaster
- recognize hands-on activities used to train response teams
- recognize how to minimize the potential damage of disruptive DiRT tests
- recognize how Windows Update works
- recognize the importance of upgrading and patching systems
- recognize the nine principles of Site Reliability Engineering
- recognize what data to include in a backup, such as full operating system, personal files, and company data
- use the Backup and Restore feature and describe how it differs from the File History feature



Network Admin

Objectives

- In this lab, you will perform Network Admin tasks such as using the Backup/Restore feature of Windows 10, using OneDrive, calculating availability metrics and identifying the benefits of a SRE team engagement. You will then identify given scenarios as events, alerts or incidents, identify incidents as symptoms or causes, identify deployment types and testing event types.

Track 2: DevOps Engineer

In this track of the Network Admin to Site Reliability Engineer Skillssoft Aspire journey, the focus will be on build & release engineering best practices, automation and simplicity best practices for SRE, then move to SRE postmortem culture best practices, and finish with cloud and container architectures for the SRE.

View Less ^

▶ 11 courses | 11h 38m 21s ▶ 1 lab | 8h



Build & Release Engineering Best Practices: Release Engineering

Objectives

- provide an overview of release engineering
- differentiate between release engineering and release management
- describe the role of a release engineer
- list the four major principles of release engineering - self-service model, high velocity, hermetic builds, and enforcement of policies and procedures
- provide an overview of the Google automated release system Rapid and describe considerations for building
- provide an overview of branching as it relates to the Rapid release model
- describe the importance of continuous testing as it relates to the Rapid release model
- provide an overview and example of software packaging/distribution and describe the importance of a packaging management system
- list the components of a Rapid system
- describe the goal of a successful deployment process
- provide an overview of configuration management
- describe the importance of starting release engineering practices from the beginning
- describe how companies can implement/incorporate release engineering



Build & Release Engineering Best Practices: Release Management

Objectives

- provide an overview of release management and recognize its key terms
- provide an overview of the release management process
- provide an overview of key concepts such as code management and version control
- describe the release management cycle and each step in the cycle
- list common challenges surrounding release management
- recognize the importance of an enterprise release management framework
- list project management principles as they relate to release management
- provide an overview of the release management processes by stages in the software development life cycle
- list common toolsets used by a release engineer in their job
- provide an overview of metrics and KPIs such as release downtime, priority of releases, and on-time releases
- describe release management and DevOps in an Agile world
- differentiate between continuous delivery, continuous deployment, and continuous integration
- provide an overview of the release deployment process



Best Practices for the SRE: Automation

Objectives

- describe automation and its benefits
- differentiate between scripts and programs as they relate to automation
- provide an overview of how automation has evolved at a company such as Google
- describe how automation consistency can add value
- describe how automation can provide a platform that can be extended and how it can centralize mistakes
- describe how automation can reduce the mean time to repair for common faults
- list examples of how system automation can react/respond quicker than human intervention
- recognize the time-saving benefits of automation
- recognize governance and reliability benefits of automation
- recognize how automation can allow you to accomplish more using fewer resources
- recognize key metrics that can be captured and reported on using automation
- list potential challenges of automation such as integration, ROI, complexity, and security
- recognize automation value for a Google SRE



Best Practices for the SRE: Use Cases for Automation

Objectives

- recognize how to use PowerShell for automation tasks in Windows
- recognize how to automate Active Directory user creation and user provisioning
- recognize how to automate tasks such as patching and Windows updating
- list Windows features that can automate aspects of software rollouts such as Windows Deployment Services and Volume Activation Management Tool
- provide an overview of setting up a bulk enrollment of devices to be managed by mobile device management
- list common Google SRE use cases for automation
- provide an overview of automation classes and describe the path the evolution of automation follows
- recognize the advantages and considerations when automating all the things
- list the benefits of applying automation to cluster turnups
- recognize how to possibly detect and resolve automation issues that relate to cluster turnups
- describe how automation processes can vary and how there can be an inclination to specialize
- recognize why reliability is the fundamental feature of automation
- provide an overview of automation with respect to enabling failure at scale and recognize possible troubleshooting and backup strategy considerations should automation fail
- recognize steps to measure the success or failure of automation



SRE Simplicity: Software System Complexity

Objectives

- recognize how simple systems and software are easy to understand, maintain, and test
- outline the importance of simple code, e.g., having fewer bugs
- define the concept of software complexity
- describe the concept of theoretical complexity
- recognize ways to determine software usage difficulty, e.g., is it functional and usable?
- determine the structural complexity of software, e.g., how difficult it is for a developer to understand and maintain software?
- identify how complicated it might be for an organization to coordinate and collaborate on software development
- differentiate between complex and complicated code
- describe the importance of measuring complexity
- list the benefits of complexity analysis
- define the concept of cyclomatic complexity
- provide an overview of the Halstead metric, used to determine levels of complexity
- provide an overview of the maintainability index, used to measure how maintainable source code is
- outline the concept of class coupling
- outline how to use NPATH to measure the complexity of a piece of code
- identify the importance of prioritizing the simplification of projects and resources



SRE Simplicity: Simple Software Systems

Objectives

- outline software development stability and how it relates to SREs
- outline software development agility and how it relates to SREs
- describe stability testing and why it's performed
- differentiate between agile metrics, such as lead-time, cycle time, team velocity, and open/close rates
- contrast mean time between failures and mean time to recovery/repair
- outline security metrics and how they relate to software quality
- recognize how to formulate a value hypothesis
- differentiate between essential complexity vs. accidental complexity and recognize how simple, "boring" software is ideal
- describe source control systems and how to deal with engineers who do not want to share code
- identify how APIs can lead to a path of simplicity
- define the "negative lines of code" metric and describe why it is important
- describe the modularity of APIs
- differentiate between simple and complex releases



SRE Postmortems: Blameless Postmortem Culture Creation

Objectives

- outline a premortem approach and describe its benefits
- recognize how to do a premortem on a project and the best practices to follow
- outline the concept of postmortem and postmortem culture
- list the goals of writing a postmortem, such as documenting the incident, root causes, and preventative actions
- describe the idea of blameless postmortems
- outline how to achieve a blameless postmortem
- list benefits of a blameless postmortem culture shift
- identify best practices for implementing a thriving postmortem culture
- recognize the value of collaboration and knowledge-sharing throughout the postmortem workflow
- describe why it is important to review postmortems
- list activities that could help introduce a new postmortem culture such as 'postmortem of the month,' 'postmortem groups,' 'clubs,' etc.
- identify how asking for feedback can increase postmortem effectiveness
- list some examples of bad postmortems and potential areas for improvement, such as missing context, omitted details, missing action items, finger-pointing, and animated language
- outline a healthy postmortem, including clarity, concrete action items, blamelessness, depth, promptness, and conciseness
- list the benefits of enforcing blameless behavior such as using blameless language
- give reasons why it is essential to reward postmortem outcomes
- outline why it is beneficial to share postmortems as widely as possible
- describe how to respond to postmortem culture failures
- list the benefits of using a postmortem template
- identify how tools can help facilitate postmortem culture



Cloud and Containers for the SRE: Cloud Architectures & Solutions

Objectives

- describe the core characteristics of Software as a Service (SaaS), name key considerations when choosing SaaS, and recognize the key advantages of using SaaS for SRE work
- define Infrastructure as a Service (IaaS), outline what makes up a typical IaaS architecture, and name advantages, limitations, usage considerations, use cases, and key providers of IaaS
- define Platform as a Service (PaaS), name the key characteristics and advantages of PaaS, give examples of providers, and describe the cost considerations of using PaaS
- describe what is meant by Container as a Service (CaaS) and "container services", name components of a container environment, and give examples of orchestration tools and container technologies
- describe what is meant by an Anything as a Service (XaaS) solution and recognize the benefits and potential drawbacks of using XaaS solutions
- define Desktop as a Service (DaaS), outline the services offered by a DaaS provider, and list the individual characteristics and advantages of popular DaaS providers
- give examples of the kinds of processes that constitute Business Process as a Service (BPaaS), describe key characteristics of BPaaS, and name vital considerations when choosing a BPaaS for SRE-related work

- outline what comprises a private cloud, recognize which cloud service models can be delivered in them, describe ways to use them, and distinguish the advantages and disadvantages of their use
- describe the key characteristics of a public cloud, recognize when to implement a public cloud model, and distinguish the advantages and disadvantages to using this model for SRE work
- describe what's meant by a hybrid cloud, name the components of hybrid cloud architecture, and distinguish the advantages and disadvantages of using it for SRE work
- describe what is meant by a community cloud, give examples of its use, and distinguish the advantages and disadvantages to using it for SRE work
- differentiate between on-premises and off-premises solutions and compare their key advantages in relation to SRE work
- describe how cloud orchestration works and why it's used, recognize its advantages and two model types, and list popular providers
- describe how cloud automation is used and why, recognize its benefits, differentiate between cloud automation and orchestration, and name popular cloud automation tools
- outline what's meant by cloud elasticity, why it's used, and how it relates to resources and costs and recognize its potential drawbacks
- describe what's meant by ubiquitous access for cloud implementation and how this relates to convergence and embedded computers
- outline how cloud computing multitenancy works, distinguish its benefits and drawbacks, recognize how it applies to public and private clouds, and define container architecture and serverless computing
- define what's meant by an on-demand self-service and list the key advantages of using these services for SRE work
- describe what's meant by cloud bursting and how it can help manage peaks in IT demand and recognize the advantages of and considerations for its use
- outline what's involved in cloud security and recognize the challenges and considerations when implementing a secure cloud solution
- differentiate among the various cloud payment models, listing the features of each
- describe the purpose of cloud resource pooling, outline how it works, and name some services that a resource pooling strategy can apply to



Cloud and Containers for the SRE: Containers

Objectives

- outline the features and purpose of containers
- differentiate between containers and virtual machines, and list the benefits of using containers, such as consistent environments, isolation, and ability to run anywhere
- outline the characteristics and purpose of Kubernetes
- describe the essential components of Kubernetes
- outline the purpose of Docker container images and the attributes of Docker Engine
- describe how the Docker Desktop can be used to containerize applications on a desktop
- recognize how to use Docker Hub to find and share container images with others
- outline how to use Docker to manage container ecosystems
- describe the characteristics of Docker Registry and recognize how it can be used to standardize application packaging and distribution
- name the benefits of using AWS to run containers
- recognize how containers enable a more efficient continuous delivery system and why they are needed for SRE
- define continuous deployment with containers
- outline continuous integration with Docker
- list the Docker storage type categories and Docker data types
- list considerations when implementing container security
- name high-availability solutions for containers, such as Docker Datacenter (DDC)
- outline the steps involved in performing live migration of containers
- describe what is meant by Containers as a Service



Cloud and Containers for the SRE: Implementing Container Solutions

Objectives

- differentiate between Git and Github and recognize how they can be used for revision control and to support collaborative development
- provide an overview of Distributed Version Control Systems (DVCS)
- outline how the Jenkins workflow tool works
- recognize how to use Jenkins Pipeline to implement and integrate continuous delivery pipelines
- differentiate between declarative and scripted pipelines
- demonstrate how to write pipeline-as-a-code using a declarative syntax with Jenkinsfiles
- illustrate code used at the 'build' stage
- illustrate code used for recording failures
- illustrate deployment code used after successful execution of the build and test stages
- set environment variables using Jenkins Pipeline
- describe key steps and factors of a code review
- use Kubernetes to deploy applications with high availability, scalability, and resilience



Final Exam: SRE DevOps Engineer

Objectives

- demonstrate how to use Kubernetes to deploy applications with high availability, scalability, and resilience
- describe continuous integration with Docker
- describe how automation can reduce the meantime to repair for common faults
- describe how automation consistency can add value
- describe the importance of continuous testing as it relates to the Rapid release model
- describe the release management cycle and each step in the cycle
- differentiate between complex and complicated code
- differentiate between containers and virtual machines and list container benefits such as consistent environment, isolation, run anywhere
- differentiate between continuous delivery, continuous deployment, and continuous integration
- differentiate between essential complexity vs. accidental complexity and recognize how simple, "boring" software is ideal
- differentiate between Git and Github and recognize how they can be used for revision control and to support the collaborative development
- differentiate between release engineering and release management
- differentiate between scripts and programs as they relate to automation
- differentiate between simple and complex releases
- distinguish between complex and complicated code
- identify how APIs can lead to a path of simplicity
- identify when to implement a public cloud model and list key advantages for SREs
- list common challenges surrounding release management
- list considerations when implementing container security
- list potential challenges of automation such as integration, ROI, complexity, and security
- list the advantages of using a hybrid cloud model and list key advantages for SREs
- list the four major principles of release engineering - self-service model, high velocity, hermetic builds, and enforcement of policies and procedures
- list the goals of writing a post-mortem such as documenting the incident, root causes, and preventative actions
- list Windows features that can automate aspects of software rollouts such as Windows Deployment Services and Volume Activation Management Tool
- outline software development agility and how it relates to SREs
- outline software development stability and how it relates to SREs
- outline the importance of simple code, e.g., having fewer bugs
- provide an overview of a premortem approach including its benefits
- provide an overview of blameless post-mortems
- provide an overview of branching as it relates to the Rapid release model
- provide an overview of containers
- provide an overview of Docker
- provide an overview of how automation has evolved at a company such as Google
- provide an overview of infrastructure as a service (IaaS) and list key advantages for SREs
- provide an overview of Kubernetes
- provide an overview of release engineering
- provide an overview of release management and recognize its key terms

- provide an overview of setting up a bulk enrollment of devices to be managed by the mobile device management
- provide an overview of software as a solution (SaaS) and list key advantages for SREs
- provide an overview of the community cloud model and list key advantages for SREs
- provide an overview of the platform as a service (PaaS) and list key advantages for SREs
- provide an overview of the private cloud model
- provide an overview of the release management process
- provide examples of a poor bad post-mortem including potential areas to improve such as missing context, omitted details, missing action items, finger-pointing, animated language
- recognize how containers enable a more efficient continuous delivery system and why they are needed for SRE
- recognize how simple systems and software are easy to understand, maintain, and test
- recognize how to automate Active Directory user creation and user provisioning
- recognize how to automate tasks such as patching and Windows updating
- recognize how to do a premortem on a project as well as best practices to follow
- recognize how to use Jenkins Pipeline to implement and integrate continuous delivery pipelines
- recognize how to use PowerShell for automation tasks in Windows
- recognize key steps and factors of a code review
- recognize the benefits of implementing a content as a service (CaaS) solution
- recognize the importance of reviewing post-mortems
- recognize the importance of sharing post-mortems as widely as possible
- recognize the value of collaboration and knowledge sharing throughout the post-mortem workflow
- recognize ways to determine software usage difficulty, e.g., is it functional and usable?
- recognize when to implement a public cloud model and list key advantages for SREs



DevOps Engineering

Objectives

- In this lab, you will perform DevOps Engineer tasks such as identifying appropriate release management roles given tasks and identifying automation opportunities, categorize software deployment as structural, computational, or organizational complexity. You will also identify scenarios that will benefit from improving lead time, cycle time, team velocity, or open/close rate. The identify the signs of a bad postmortem, identify examples of SaaS, PaaS, and IaaS. Finally, identify whether container data, data volumes, data volume containers, or external storage would be most appropriate for given container scenarios and create a simple JenkinsFile.

Aspire Journeys: Network Admin to Site Reliability Engineer

Track 3: Chaos Engineer

In this track of the Network Admin to Site Reliability Engineer Skillsoft Aspire journey, the focus will be on troubleshooting, emergency response and incident handling, testing for reliability, load balancing, overload and cascading failures, distributed reliability, data pipelines and integrity, and deploying products at scale.

[View Less](#) ^

17 courses | 18h 20m 12s | 1 lab | 8h



Sven Batalla
Product Owner/Scrum Master

SRE Troubleshooting Processes

Objectives

- describe how engineers think differently to "novices" when it comes to troubleshooting
- outline best practices and approaches to troubleshooting and how to keep those skills sharp
- outline an idealized troubleshooting model (e.g., report, triage, examine, diagnose, test/treat, and cure.)
- list potential pitfalls to avoid, such as looking for symptoms that are not relevant
- outline how to manage operational loads
- recognize the importance of an adequate initial problem report
- recognize the importance of triaging problems from the onset
- recognize the importance of examining each component of a system to understand whether it is functioning properly
- identify the steps and approaches used to diagnose issues
- describe methods for testing and treating possible causes to identify actual problems
- recognize how to simplify and reduce troubleshooting using techniques such as dividing and conquering
- describe the "what, why, where" technique and how it can be used to diagnose a malfunctioning system
- interpret how determining who last touched a system can be helpful when identifying what is going on with a system
- define what is meant by "negative results"
- recognize that systems are complex and that often you can only identify probable cause factors to document what went wrong with a system
- outline steps to make troubleshooting easier



SRE Troubleshooting: Tools

Objectives

- outline the process and purpose of logging and name the benefits of text logs
- describe the characteristics and purpose of whitebox monitoring
- describe the characteristics and purpose of blackbox monitoring
- access and navigate the Windows Event Viewer
- open the System Information panel in Windows and use it to view and collect system information
- use Windows Resource Monitor to display real-time hardware and software usage information
- summarize the characteristics of Dapper and outline how it can be used to troubleshoot a distributed system
- process logs using the Google Cloud Dataflow workflow tool
- recognize how the StatsD standard is used for instrumenting software and exposing metrics
- outline the characteristics, components, and purpose of the Prometheus open source systems monitoring and alerting toolkit
- outline how to manually send a request to the `/api/search` endpoint to identify failures



SRE Emergency & Incident Response: Responding to Emergencies

Objectives

- outline the fundamental emergency response principles SREs need to be familiar with and recognize the critical steps to take when a system breaks
- recognize the benefits of performing test-induced emergencies and outline what this involves
- name the causes and outcomes of change-induced emergencies and outline how to respond to these emergencies
- define what is meant by a process-induced emergency, describe the effects of them, and outline how to respond to them
- describe why it is vital to keep a history of outages and mistakes and outline best practices when doing so
- recognize the importance of asking important, relevant, and challenging questions
- define what is meant by proactive testing, compare it to reactive testing, recognize the importance of encouraging proactive testing, and name best practices when carrying out this type of testing
- define what is meant by business continuity and describe why this type of planning matters
- outline the six steps involved in developing a business continuity plan
- outline methods to test a business continuity plan, recognize the importance of testing this type of plan, and describe some tips when testing
- recognize the importance of ongoing efforts to review and improve a business continuity plan and outline how to go about doing it
- recognize the importance of having 'top-level' support for business plans and promoting user awareness, and outline how to achieve these goals
- define what is meant by a business impact analysis, outline how to conduct one and its typical structure, and name the possible effects on business operations
- recognize the importance of developing an IT disaster recovery plan, list the goals of this type of plan, and describe what to consider when developing one
- outline key steps to creating a working disaster recovery plan
- name some types of IT recovery strategies and recognize the importance of recovery strategies developed for IT systems, applications, and data



SRE Emergency & Incident Response: Incident Response

Objectives

- summarize the requirements, goals, best practices, job roles, and tools involved in managing and responding to incidents
- recognize the importance of incident response planning and the characteristics of incidence response plans
- describe what is meant by each one of the 'three Cs' of incident management (coordinate, communicate, and control)
- restate the duties of the prominent job roles involved in incident response (Incident Commander, Communications Lead, and Operations Lead) as well as those of other, supporting roles
- recognize the best practices for handling managed incidents
- recognize the best practices for handling unmanaged incidents
- describe why correctly declaring and classifying incidents is essential and when these activities should be carried out
- assess why incident response training is necessary and list the responsibilities of an incident response analyst
- outline tips and strategies for incident response preparation
- outline how to go about practicing incident management processes and describe why these activities are important
- outline how to use incident management data for measuring team performance and identify the KPIs to look out for
- differentiate between the standard tools used for incident response
- judge when to create a CSIRT and choose who should be on that team
- identify the different purposes of the various roles on a CSIRT
- describe the elements of an incident response policy



SRE Testing Tasks: Software Reliability & Testing

Objectives

- outline what's involved in reliability testing and describe testing techniques, such as unit, integration, system, production, stress, and rollouts entangle tests
- list standard factors that can influence software reliability
- describe why SREs might carry out reliability testing
- name and describe some common SRE metrics
- describe the features and benefits of the mean time to failure (MTTF) metric and outline how to use it in SRE work
- define the mean time to respond (MTTR) metric and describe why it might be used in SRE
- define the mean time to resolve (MTTR) metric and outline when and how to use it for SRE work
- define the mean time between failures (MTBF) metric and outline when and how to use it for SRE work
- describe what's involved in software unit testing for SRE work, including when it's performed, who performs it, and the tasks involved
- define integration testing as it applies to SRE, list three associated method types, and outline how to perform an integration test, detailing the tasks involved
- outline what's involved in system testing in SRE, when it is performed, and who performs it
- outline what's involved in acceptance testing for SRE, when it's typically performed, and who performs it
- outline what's involved in production testing for SRE and recognize its purpose
- outline how to carry out configuration testing in SRE work and name the pre-requisites and objectives of this type of testing
- describe how and when to perform a stress test for SRE work
- define a canary test and outline what's involved in carrying out these types of tests in SRE work



SRE Testing Tasks: Testing Considerations

Objectives

- recognize how to create an effective test and build environment
- outline the procedures involved in software testing at scale
- recognize the importance of testing SRE-developed tools
- describe how to create barrier defenses against risky software
- recognize the challenges of testing for disasters
- outline when and how to use statistical testing
- define informal deadlines for test results and recognize when to use them
- recognize key considerations when pushing to production
- describe why it's best to expect certain tests to fail
- outline what an emergency break glass procedure is and in what situations it's used
- recognize the importance of integration testing configuration files
- describe what fake back-end versions are used for and where they factor in during release tests and monitoring



SRE Load Balancing Techniques: Front- end Load Balancing

Objectives

- define what is meant by front-end load balancing, recognize how it improves performance, classify load balancer types, and describe three load balancer algorithms
- list considerations when implementing load balancing and outline several techniques to achieve it
- name possible uses of the concepts associated with front-end load balancing
- outline how to balance loads using DNS load balancers
- outline how to balance loads using virtual IP load balancers
- describe how load balancing should be performed if working with virtualization, the cloud, and containers
- describe the features of load balancers and their use in application delivery and security
- outline methods for managing and handling overload
- relate how the Google Front End Service is used to manage loads
- indicate how the Andromeda Virtualization Stack is used as a software-defined network (SDN)
- describe the architecture and components of the Maglev network load balancing server and how it's used for high availability
- relate how the Envoy edge and service proxy works and recognize the benefits of its use



SRE Load Balancing Techniques: Data Center Load Balancing

Objectives

- describe the various data center load balancing techniques and how they increase performance
- describe what is meant by a load balancing policy and how it's applied to load balancers
- outline a simple approach to dealing with unhealthy tasks using flow control
- list various tips and tricks for working with and optimizing load balancing
- describe methods for limiting the connection pool with subsetting
- name and describe the various components that make up load balancing
- outline how loads can be balanced using Internal HTTPS Load Balancing
- outline how loads can be balanced using External HTTPS Load Balancing
- outline how loads can be balanced using Internal TCP/UDP Load Balancing
- outline how loads can be balanced using External TCP/UDP Load Balancing
- outline how loads can be balanced using SSL Proxy Load Balancing
- outline how loads can be balanced using TCP Proxy Load Balancing



Sven Batalla
Product Owner/Scrum Master

Site Reliability Engineer: Managing Overloads

Objectives

- define what is meant by operational loads, list their types, and describe how they relate to optimal performance
- outline the purpose of pages and how to manage them
- recognize the benefits of using tickets
- outline the activities involved in ongoing operational responsibilities
- identify how operational overload occurs and name considerations related to operational threshold
- outline steps to mitigate overloads
- list the potential consequences of overloads, including serious illness to staff
- recognize the importance of prioritizing work and tasks
- recognize the pitfalls of the queries per second metric
- name capacity options, such as per customer limitations
- recognize the benefits of client-side throttling
- define the concept of criticality, name four criticality values, and identify the purpose of criticality and each value
- describe the purpose and characteristics of utilization signals
- outline processes for working with overload errors
- describe mechanisms available to avoid retrying requests, such as per-request retry budget and per-client retry budget
- outline how counters can help prevent overloads
- describe how loads from connections can help recognize and prevent overloads
- identify potential problems caused by new connection bursts



Sven Batalla
Product Owner/Scrum Master

Site Reliability Engineer: Managing Cascading Failures

Objectives

- define what is meant by cascading failures and identify situations in which this term is used
- describe how server overloads can lead to cascading failures
- define what is meant by resource exhaustion and describe its consequences
- list CPU considerations as they relate to failures and overutilization
- list factors that can contribute to memory exhaustion
- recognize how file descriptors and threads can directly lead to failures
- recognize how resource exhaustion can travel from one resource to another
- recognize how resource exhaustion can lead to service unavailability
- outline how to prevent server overloads
- outline steps to ensure efficient queue management
- differentiate between load shedding and graceful degradation
- define what is meant by code retries and recognize why it is relevant to the topic of cascading failures
- recognize the benefits of setting deadlines
- recognize how propagating cancellations can reduce unneeded work
- define what is meant by latency considerations, including bimodal latency, and describe how to address this class of problems
- outline the steps involved in managing slow startups and working with cold caching
- differentiate between the various cascading failure triggers
- outline how to test cascading failures
- list steps to immediately address cascading failures



Distributed Reliability: SRE Critical State Management

Objectives

- describe critical state management and how it applies to distributed systems and affects reliability
- define the CAP theorem and describe how it relates to distributed systems
- outline how to coordinate system failures on distributed systems
- differentiate deterministic and nondeterministic algorithms and how they relate to distributed systems
- describe the system models that can be used with distributed systems
- define the concept of distributed consensus and list the stages of validation
- define the concept of Byzantine fault and describe how it applies to distributed systems
- describe the distributed consensus architecture patterns used in distributed systems
- describe best practice and tricks for increasing performance for distributed systems
- define the Multi-Paxos protocol and describe how it relates to distributed systems
- outline how to deploy distributed consensus-based systems and name some key considerations
- name and describe the key considerations when monitoring distributed consensus systems



Distributed Reliability: SRE Distributed Periodic Scheduling

Objectives

- describe what is meant by cron and how to use it for scheduling jobs
- outline the purpose of cron jobs, name their components and operators, and give examples of their use
- describe the characteristics of the cron syntax and provide syntax examples
- define the concept of idempotency and outline how distributed periodic scheduling works with cron
- outline how to scale cron for large-scale system deployments
- describe how the PAXOS distributed consensus algorithm works
- outline how replication is performed using the PAXOS algorithm
- recognize best practices when using the PAXOS algorithm
- describe how Google approached building cron
- outline how to perform distributed scheduling
- demonstrate how to schedule cron jobs
- illustrate how to use tools to generate crontab syntax



SRE Data Pipelines & Integrity: Data Pipelines

Objectives

- describe the characteristics of and rationale for using data processing pipelines
- recognize characteristics of the Extract Transform Load (ETL) pipeline model
- define business intelligence and data analytics in the context of data processing and give an example data analytics use case
- list characteristics of machine learning (ML) applications
- define what is meant by service-level objectives (SLOs) and describe how they relate to pipeline data
- outline how to plan for dependency failures
- recognize how to create and maintain pipeline documentation
- outline the stages of a typical development lifecycle
- describe how to reduce hotspotting
- recognize how to implement autoscaling to handle spikes in workloads
- describe how to adhere best to access control and security policies
- plan escalation paths that ensure quick and proactive communication
- describe the effect big data can have on simple pipeline patterns
- list the challenges with using the periodic pipeline pattern
- describe the issues that can occur due to uneven work distribution
- list the potential drawbacks of periodic pipelines in distributed environments
- describe what comprises Google Workflow and outline how it works
- outline the stages of execution in Google Workflow, describing what they entail
- recognize the key factors to ensuring business continuity in big data pipelines using Google Workflow



SRE Data Pipelines & Integrity: Pipeline Design

Objectives

- differentiate between the various pipeline features
- differentiate between idempotent and two-phase mutations
- describe what the checkpointing technique is used for, in what situations it's used, and the advantages of its use
- list common code patterns, such as reusing code and the microservice approach
- describe the purpose and characteristics of the maturity matrix
- list example milestones for the failure tolerance maturity matrix characteristic
- list examples of the scalability maturity matrix characteristic
- list examples of the monitoring and debugging maturity matrix characteristic
- list examples of the implementation maturity matrix characteristic
- list examples of the testing maturity matrix characteristic
- differentiate between the different potential failure modes
- list potential causes of an outage
- recognize event delivery system design and architecture considerations
- outline how event delivery system operations are used using practical examples
- describe considerations when planning for customer integration and support



SRE Data Pipelines & Integrity: Data Integrity

Objectives

- define data integrity and recognize how it relates to site reliability engineering and user data, its associated risks, and how to mitigate them
- list vital data integrity requirements and considerations and differentiate between data integrity and security
- outline how to choose a strategy for data integrity
- define what is meant by SRE objectives and outline how they can help maintain data integrity and ensure data availability
- differentiate between different types of failures resulting in data loss
- recognize challenges for maintaining data integrity
- state what is meant by combinations of data integrity failure modes and the multiple layers of defense in depth
- define the soft deletion layer of defense-in-depth and recognize the benefits of a soft deletion strategy
- name and describe some important considerations when implementing backup and recovery strategies
- define what is meant by the early detection layer of defense-in-depth and describe how it relates to data loss prevention
- list some common challenges faced by cloud developers
- describe what comprises out-of-band data validation and recognize some important considerations when implementing this strategy
- recognize the importance of setting up systems and knowing that data recovery will be successful
- name and define some of the core principles of SRE and outline some best practices for applying these



SRE Products at Scale: Product Launches

Objectives

- describe what's involved in a product or feature launch
- outline what's involved in launch coordination engineering
- list the specific role requirements and responsibilities of a launch coordination engineer
- list criteria to meet for a successful launch process
- recognize what's contained in a launch checklist and outline best practices for developing one
- outline the characteristics and benefits of gradual and staged rollouts
- outline the framework of a feature flag and give some examples of feature flags
- outline how to deal with abusive or upset clients
- recognize the consequences of various overload situations and outline strategies to prevent them
- list some problems launch coordination engineers may be unable to resolve and give some tips for their resolution
- describe what's involved in a simple production readiness review (PRR) model and how to set up, analyze, and improve the PRR process
- recognize the various aspects of the SRE engagement model, including support, documentation, and consultation
- outline the phases, potential candidates, and benefits of the early engagement model
- recognize the characteristics and benefits of SRE engagement frameworks



Final Exam: Chaos Engineer

Objectives

- describe challenges for maintaining data integrity
- describe CRON and how to use it for scheduling jobs
- describe CRON Jobs and its components
- describe deterministic and non-deterministic algorithms and how they relate to distributed systems
- describe frontend load balancing and the importance of using them to increase performance
- describe how engineers think differently to "novices" when it comes to troubleshooting
- describe how load balancing needs to be performed taking into consideration virtualization, the cloud, and containers
- describe how loads can be balanced using External HTTPS Load Balancing
- describe how loads can be balanced using SSL Proxy Load Balancing
- describe how loads can be balanced using TCP Proxy Load Balancing
- describe how server overloads can lead to cascading failure
- describe load balancing techniques and algorithms
- describe operational loads and how they related to optimal performance
- describe steps to ensure efficient queue management
- describe the benefits of client-side throttling
- describe the characteristics and purpose of blackbox monitoring
- describe the characteristics and purpose of whitebox monitoring
- describe the CRON syntax and provide syntax examples
- describe the importance of incident response training
- describe the mean time between failures metric
- describe the meantime to respond metric
- describe the system models that can be used with distributed systems
- describe when to use acceptance testing
- differentiate between idempotent and two-phase mutations
- differentiate between the various pipeline features
- discuss software testing at scale
- identify the system models that can be used with distributed systems
- list characteristics of machine learning (ML) applications
- list CPU considerations as it relates to failures and overutilization
- list data integrity requirements
- list potential pitfalls to avoid, such as looking for symptoms that are not relevant
- list the main roles in incident response (Incident Commander, Communications Lead, Operations Lead)
- outline an idealized troubleshooting model (e.g., report, triage, examine, diagnose, test/treat, and cure.)
- outline best practices and approaches to troubleshooting and how to keep those skills sharp
- outline the benefits of using tickets
- outline the process and purpose of logging and name the benefits of text logs
- provide a general overview of the six steps involved in developing a plan
- provide an overview of a typical development lifecycle
- provide an overview of backup and recovery methods
- provide an overview of business continuity and describe why business continuity planning matters

- provide an overview of data integrity
- provide an overview of Google Workflow
- provide an overview of key principles SREs need to be familiar with for emergency response and recognize key steps to take when a system breaks
- provide an overview of pages
- provide an overview of resources exhaustion
- provide an overview of the checkpointing technique
- provide an overview of the maturity matrix
- provide an overview of the meantime to failure metric
- provide an overview of the production readiness review process
- recognize aspects of the SRE engagement model
- recognize best practices for handling unmanaged incidents
- recognize how to create and maintain pipeline documentation
- recognize how to create a test and build an environment
- recognize how to develop a launch checklist
- recognize how to identify cascading failures
- recognize key factors to ensuring business continuity
- recognize the importance of encouraging proactive testing
- recognize the importance of incident response planning
- recognize the importance of testing SRE-developed tools
- recognize the pitfalls of the queries per second metric



Chaos Engineer

Objectives

- In this lab, you will perform tasks such as using Resource Monitor, and identifying testing types, effective incident management, load balancing types and algorithm types. Then, create CRON statements, identify failure scenario characteristics and match launch coordination challenges with mitigation techniques.

Aspire Journeys: Network Admin to Site Reliability Engineer

Track 4: Site Reliability Engineer

In this track of the Network Admin to Site Reliability Engineer Skillssoft Aspire journey, the focus will be on scaling the SRE team, operational loads, communication and collaboration, managing software reliability metrics, and the SRE engagement model.

9 courses | 8h 51m 46s 1 lab | 8h



Sven Batalla
Product Owner/Scrum Master

SRE Team Management: Scaling the Team

Objectives

- describe the steps that should be followed when onboarding a new site reliability engineer
- outline the process and best practices for onboarding a new site reliability engineer
- outline the process for onboarding a site reliability engineer remotely
- describe the training patterns and tactics for onboarding a site reliability engineer
- name the tools that can be used for onboarding a new site reliability engineer
- recognize the importance of creating an environment that focuses on structured and well-planned learning
- outline the best constructive learning paths for training a new site reliability engineer
- describe the technical skills required for managing a production service
- identify the skills and best practices used for reverse engineering a production service call
- outline the training that a software reliability engineer needs to undertake to perform on-call support
- outline the best practices that a software reliability engineer should follow when performing on-call support
- list the skills that a software reliability engineer needs to acquire to perform on-call support



SRE Team
Management:
Managing
Operational Loads

Objectives

- describe what is meant by operational load and outline the three general categories of operational load
- outline how on-call engineers depend on pages to respond to incidents and outages
- outline the steps involved in responding to emergency incidents
- outline the purpose of customer request support tickets and provide examples of simple and complex tickets
- describe the essential components of a typical ticketing system
- recognize how to use service level objectives (SLO) to ensure timely responses and resolutions
- describe what is meant by toil and provide examples of toil, such as applying schema changes to a database
- differentiate between types of toil including automated, manual, repetitive, and tactical
- outline steps to track and identify toil and describe why less toil is better
- describe how to measure and calculate toil
- outline steps to minimize or eliminate toil completely
- differentiate between toil and complexity and describe approaches to address complexity
- describe how toil can negatively effect staff including through low morale and confusion amongst SREs
- list key metrics used for managing interrupts, such as the severity of the interrupt
- outline human element factors to consider when dealing with interrupts, such as distractibility
- summarize the key concepts covered in this course



SRE Team
Management:
Operational
Overload

Objectives

- describe the term ops mode and differentiate between ops mode and nonlinear scaling
- outline factors that contribute to team morale and stress such as financial and managerial impacts
- list the details to include in an IT emergency plan
- outline possible emergencies to plan for, such as undiagnosed alerts and knowledge gaps
- describe how knowledge sharing can help teams plan for emergencies and recover from failures
- recognize key factors of a high-quality postmortem
- classify team emergencies into either 'toil' or 'not toil' categories
- recognize the importance of service level objectives (SLOs) as they relate to a long-term SRE focus
- describe steps to ensure a team-first approach to fixing overload issues
- outline the importance of properly explaining findings and applying an appropriate level of detail for explanations
- list key attributes of successful teams including purpose, trust, and awareness
- differentiate between questioning techniques such as open/closed, funnel, probing, and leading



Core Skills for Site Reliability Engineers: SRE Collaboration & Communication

Objectives

- describe the importance of communication and collaboration and running effective SRE meetings
- describe best practices when running meetings for SREs
- recognize the importance of communication and collaboration and running effective SRE meetings virtually
- describe the best practices when running virtual meetings for SREs
- list the types of meetings that are useful for an SRE
- describe how pair programming can be an effective collaboration tool for an SRE
- describe how an SRE can work effectively and collaboratively with distributed teams
- outline best methods for collaborating with external customers
- outline methods for working with external customers to develop and support applications
- describe real-life scenarios in which effective collaboration has helped external customers resolve their technical support issues
- state the types of tools that can be used to develop software with external customers rapidly
- compare the features of various conferencing software that can be used for virtual meetings



SRE Metric Management: Software Reliability Metrics

Objectives

- describe useful SRE data analysis metrics and how they can be effectively used to monitor and control a project
- identify some of the pitfalls encountered when using software project metrics and outline how to avoid them
- identify how to select meaningful software project metrics and recognize why some metrics have minimal value
- describe the benefits of using software metrics and outline how to monitor and track them
- name some of the best metric monitoring APIs
- outline how Azure API Management can be used for deploying and hosting API web services
- identify the metrics to track when performing CI/CD
- define manual testing and describe how it compares to automated testing
- define automated testing and describe the automated testing process and how the scope of automated testing is defined
- determine the best test cases to automate and how and when to test
- list the different types of automated testing including web applications, mobile devices, web service, and data testing
- outline the automated testing framework and automation tool best practices
- describe types of automated testing and state the goal of each test type
- compare automated and manual testing and classify the pros and cons of each method
- state reasons why automated testing may not catch all risks



**SRE Metric
Management:
Software Reliability
Monitoring and
Reporting**

Objectives

- outline various methods for analyzing the effects of faults in a system
- outline how to use fault tree analysis to determine the cause of faults in a system
- name the tools that can be used to perform fault tree analysis
- outline how to classify software defects
- describe the various types of software bugs and recognize why they occur
- differentiate between the severity and priority of software bugs
- outline best practice when defining API monitoring strategies
- state the key characteristics of API monitoring strategies
- list API monitoring tools and their strengths and weaknesses
- identify the components of the ELK Stack and how they work together for data reporting
- describe the features and benefits of Elasticsearch for storing log data
- describe the features and benefits of Kibana for viewing data
- describe the features and benefits of Beats for data collection
- describe the features and benefits of Logstash for data processing
- outline how to use Elasticsearch notifications to notify staff when API services have issues



**SRE Engagement:
Production
Readiness Review**

Objectives

- define the Production Readiness Review (PRR) model and how it's used to identify the reliability needs of a service
- describe the engagement phase of Production Readiness Review (PRR)
- describe the analysis phase of Production Readiness Review (PRR)
- describe the refactoring phase of Production Readiness Review (PRR)
- describe the training phase of Production Readiness Review (PRR)
- describe the onboarding phase of Production Readiness Review (PRR)
- describe the continuous improvement phase of Production Readiness Review (PRR)
- outline how early engagement can be used to evolve the Simple PRR model
- outline how SRE platforms and frameworks can be used to provide structural solutions
- define the SRE engagement model and outline how to use it to manage software projects
- state the various aspects of the SRE engagement model
- define the System Development Life Cycle (SDLC) and differentiate it from the SRE engagement model



SRE Engagement: The SRE Engagement Model

Objectives

- describe the SRE service lifecycle and how it compares to the traditional software development life cycle
- identify the level of SRE engagement involved during the service life cycle
- outline the steps for setting up and building SRE service relationships
- outline a roadmap for SRE sprints and communication
- outline how to measure the impact of SRE engagement
- describe how to set ground rules for SRE teams
- describe how SRE teams perform planning and execution
- describe the keys to sustaining effective ongoing relationship with other SREs and developers
- state the steps to take for scaling SRE to larger environments
- state the steps to take for ending SRE engagements
- describe the results of a case study in which the SRE engagement model was used to analyze the Google Abuse SRE and Common Abuse Tool (CAT)
- describe the results of a case study in which an SRE engagement model was used to analyze the data analysis pipeline



Site Reliability Engineer

Objectives

- In this lab, you will perform tasks such as identifying reverse engineering techniques and incident severity, identifying SRE engagement and virtual meeting best practices, and identifying the appropriate use of CI/CD metrics. Then, determine fault tree outcomes, identify incident probability and impact, and calculate availability.



Final Exam: Site Reliability Engineer

Objectives

- describe available conferencing software that can be used for virtual meetings
- describe how early engagement can be used to evolve the Simple PRR model
- describe how knowledge sharing can help teams plan for emergencies and recover from failures
- describe how pair programming can be an effective collaboration tool for an SRE
- describe how SRE teams perform planning and execution
- describe how to classify software defects
- describe how to measure and calculate toil
- describe how to measure the impact of the SRE engagement
- describe how to set ground rules for SRE teams
- describe manual testing and how it compares to automated testing
- describe the API Monitoring tools and their strengths and weaknesses
- describe the benefits of using software metrics and how to monitor and track them
- describe the best practice when defining API Monitoring Strategies
- describe the best practice when running meetings for SREs
- describe the components on the ELK Stack and how it works together
- describe the different types of automated testing, including web applications, mobile devices, web service and data testing
- describe the engagement phase of Production Readiness Review (PRR)
- describe the importance of communication and collaboration and running effective meetings
- describe the importance of communication and collaboration and running effective virtual meetings

- describe the keys to sustaining an effective ongoing relationship with other SREs and developers
- describe the level of SRE engagement during the service life cycle
- describe the onboarding phase of Production Readiness Review (PRR)
- describe the Production Readiness Review (PRR) and how it is used to identify the reliability needs of a service
- describe the purpose and benefits of Elasticsearch
- describe the SRE engagement model and how to use it to manage software projects
- describe the SRE service lifecycle and how it compares to the traditional software development life cycle
- describe the steps for ending the SRE engagements
- describe the steps that should be followed when onboarding a new site reliability engineer
- describe the System Development Life Cycle (SDLC) and how it compares to the SRE engagement model
- describe the technical skills required for managing a production service
- describe the term ops mode and differentiate between ops mode and nonlinear scaling
- describe the tools that can be used to perform fault tree analysis
- describe the training patterns and tactics for onboarding a site reliability engineer
- describe the types of meetings that useful for an SRE
- describe the various types of software bugs and why they occur
- describe types of automated testing and goal of each test type
- describe useful SRE data analysis metrics, and they can be effectively used to monitor and control the project
- describe what is meant by operational load and outline the three general categories of operational load
- describe what is meant by toil and provide examples of toil, such as applying schema changes to a database
- differentiate between types of toil, including; automated, manual, repetitive, and tactical
- explain how to classify software defects
- identify available conferencing software that can be used for virtual meetings
- identify factors that contribute to team moral and stress such as financial and managerial impact
- identify how to select meaningful software project metrics and understand why some metrics have minimal value
- identify key factors of a high-quality post-mortem
- identify some of the pitfalls encountered when using software project metrics and how to avoid them
- Identify the metrics to track when performing CI/CD
- identify the skills and best practices used for reverse engineering a production service call
- list details to include in an IT emergency plan
- list the skills that a software reliability engineer needs to acquire to perform on-call support
- name the tools that can be used for onboarding a new site reliability engineer
- outline factors that contribute to team moral and stress such as financial and managerial impact

- outline how on-call engineers depend on pages to respond to incidents and outages
- outline steps to track and identify toil and describe why less toil is better
- outline the process and best practices for onboarding a new site reliability engineer
- outline the purpose of customer request support tickets and provide examples of simple and complex tickets
- outline the steps involved in responding to emergency incidents
- recognize how to use service level objectives (SLO) ensure timely responses and resolutions
- recognize key factors of a high-quality post-mortem
- Simple PRR Model: Early Engagement

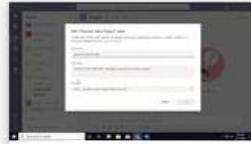
Productivity Tools for Site Reliability Engineers Optional



COURSE

Getting to know the application

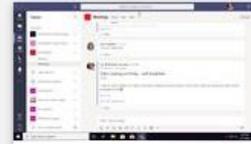
991



COURSE

Using Teams & Channels

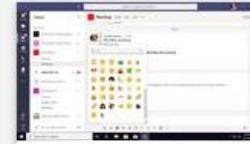
739



COURSE

Communicating via the App

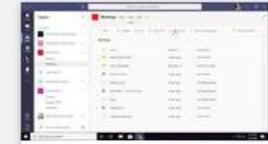
715



COURSE

Formatting, Illustrating & Reacting to Messages

535



COURSE

Creating, Finding & Organizing Files

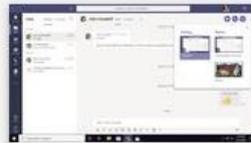
529



COURSE

Working with Apps, Tabs & Wiki

460



COURSE

Making calls, Organizing Contacts & Using Voicemail

448



COURSE

Creating, Joining & Managing Meetings

457



COURSE

Signing in & Setting Up Slack

23



COURSE

Using Channels in Slack

11



COURSE

Using Private Messaging & Communication Tools in...

13



COURSE

Creating, Finding & Sharing Information in Slack

6



COURSE

Configuring Slack

4



COURSE

Creating & Setting Up Projects in Jira Cloud

147



COURSE

Configuring & Managing Boards in Jira Cloud

97



COURSE

Planning & Working on a Software Project in Jira...

74



COURSE

Reporting in Jira Software

72



COURSE

Signing in & Navigating within Spaces

39



COURSE

Setting Up & Managing Spaces

32



COURSE

Working with Space

27



COURSE

Working with Team Members

74



COURSE

Configuring Spaces

19

Business & Leadership for Site Reliability Engineers Optional



COURSE

Clarity and Conciseness in Business Writing

1036



COURSE

Facilitating Sustainable Change

328



COURSE

Coaching Techniques That Inspire Coachees to Action

544



COURSE

Strategies for Managing Technical Teams

202



COURSE

Personal Power and Credibility

744



COURSE

Developing and Supporting an Agile Mindset

942



COURSE

Managing a Project to Minimize Risk and Maximiz...

528



COURSE

ITIL® Continual Service Improvement

171



COURSE

Building a Culture of Design Thinking

419



COURSE

Building the Foundation for an Effective Team

542



COURSE

Leading a Cross-functional Team

180

Bookshelf Optional



BOOK

Networking Fundamentals

10



BOOK

Networking All-in-One for Dummies, 7th Edition

17



BOOK

Applying Integration Techniques and Methods in...

1



BOOK

Building Dependable Distributed Systems

3



BOOK

Large Scale Network-Centric Distributed Systems

1



BOOK

Beginning Build and Release Management with TFS 201...

1



BOOK

Cloud Computing Basics

86



BOOK

Reliability Prediction and Testing Textbook

2

FOLLOW US ON:



www.skilltech.pl

email: biuro@skilltech.pl

tel. +48 22 44 88 827

SkillTech
Technology hired for excellence