ENTERPRISE

UDACITY

SCHOOL OF CLOUD COMPUTING

Site Reliability Engineer

Nanodegree Program Syllabus

Overview

The goal of this program is to equip software developers with the engineering and operational skills required to build automation tools and responses that ensure designed solutions respond to non-functional requirements such as availability, performance, security, and maintainability. The content will focus on both designing systems to automate response to issues with software sites as well as how to respond to common on-call situations.

Learning Objectives

A graduate of this program will be able to:

- Use proactive and reactive SRE strategies (monitoring, postmortem, team building, etc.) to identify reliability risks through evaluating systems and processes.
- Develop customer-centric SLOs (such as percentile targets for availability, latency, and correctness) and set up corresponding monitoring and risk mitigation measures to ensure customer happiness.
- Create and deploy automated self-healing architectures and other technologies to make the environment more maintainable.
- Design and implement organizational processes and culture that enhance product reliability, including outage/postmortem review, quarterly state of production presentation, and production readiness review.

Program information

Estimated Time

4 months at 10hrs/week*

Skill Level

Intermediate

Prerequisites

A well-prepared learner should be able to:

- Write basic functions in an object-oriented language (Python or Java), such as for loops, conditionals, control flow, Python methods, Java methods, etc.
- Write basic shell scripts in Bash or Powershell, which could include for loops, conditionals, scripting, etc.
- Understand Linux command-line (bash/shell) and UNIX shell.
- Create simple SQL queries using SELECT, JOINS, GROUP BY functions.
- Exercise networking skills including knowledge of virtual networks, DNS, subnets, and basic network troubleshooting techniques.
- Perform DevOps tasks, such as setting up monitoring, doing feature rollout, and troubleshooting production issues (ideally for large systems).
- Work with Kubernetes and basic kubectl, such as kubectl apply, kubectl create, kubectl config.

Required Hardware/Software

There are no software and version requirements to complete this Nanodegree program. All coursework and projects can be completed via Student Workspaces in the Udacity online classroom.

*The length of this program is an estimation of total hours the average student may take to complete all required coursework, including lecture and project time. If you spend about 5-10 hours per week working through the program, you should finish within the time provided. Actual hours may vary.

Course 1

Foundations of Observability

In this course learners will focus on what observability requires in terms of people and tools. To begin with, we will introduce SRE, its roles and responsibilities, and how those differ from other teams (DevOps, SysAdmin, Development). Once learners establish that, they will see how SRE helps an enterprise improve and discuss the costs associated with SRE. Learners will come to know the types of members of the SRE team, then end with the tool set that an SRE team may use to be successful.

Course Project

Observing Cloud Resources

Learners will configure a monitoring software stack to college and display a variety of metrics for commonly used cloud resources including VM scale sets, Kubernetes service, and VMs. Additionally, learners will establish and configure rules for alerting and set parameters to be notified prior to the occurrence of failures within the aforementioned cloud resources. Learners will also have the opportunity to test and observe their own implementation of the monitoring software stack to apply and showcase SRE methodologies and practices which can be transferred to real-world scenarios.

Lesson 1

SRE Roles & Responsibilities

- Identify the formation of SRE in the industry.
- Compare the SRE scope of work and functions vs. adjacent roles (DevOps, sys admin, and developers).
- Explain core skills of SRE.

Lesson 2

Improving Enterprise Workflows

- Identify common SRE practices incident response playbooks.
- Explore enterprise workflows that can use reliability engineering.
- Perform cost-benefit analysis of impact of SRE best practice on identified enterprise workflow for improvement.

Lesson 3

SRE Teams

- Illustrate collaboration best practices with development team cross-functional collaboration.
- Define the SRE team.
- Develop governance of SRE team work quality.

Lesson 4

Monitoring System Performance

- Install Prometheus/Grafana: Understand the installation steps and out-of-the-box configuration.
- Create a dashboard for host metrics (latency, errors, resource utilization CPU/RAM Disk I/O), observability dashboards, and site reliability metrics.
- Install and configure a synthetic monitoring solution.
- Create alerts for application (availability, latency) metrics, monitor an endpoint, and trigger an alert if the endpoint is down.

Course 2

Planning for High Availability & Incident Response

This course will cover monitoring, high availability (HA) and disaster recovery (DR), infrastructure as code, and database recovery and availability. We start by defining SLOs and SLIs. We then take those SLOs and SLIs and translate them into queries for Prometheus and graphs in Grafana. Next, we look at our infrastructure overview, improve it with HA principles, and then craft a DR plan. We then take that plan and deploy it via Terraform to multiple AWS regions. We wrap up the content by designing and deploying highly available databases to AWS via Terraform.

Course Project

Deploying HA Infrastructure

In this project, learners will design and deploy HA infrastructure through Terraform and deploy it to AWS. They will start by defining SLOs and SLIs and create a dashboard in Grafana for those objectives. Next, they will create a disaster recovery plan and define their high-availability infrastructure. Learners will take what they build and form Terraform code to deploy the infrastructure to multiple AWS regions. Finally, they will deploy replicated databases through Terraform code to AWS.

Lesson 1

SLOs & SLIs

- Understand what SLI/SLOs are and how each relates to an SLA.
- Define customer-centric SLOs.
- Establish a plan on how to obtain metrics for SLOs/SLIs.
- Create SLI/SLO dashboards in Grafana which display these metrics in a way that can be consumed by non-technical personnel.

Lesson 2

IT Assets, Availability & Disaster Recovery

- Determine the purpose and needs of each IT asset.
- Define a plan to consolidate IT assets.
- Create a plan to allow for high availability by selecting optimal server geography and communication.
- Create a disaster recovery plan based on a designed high-availability environment.

Lesson 3

Create & Deploy HA & DR Infrastructure Using Terraform

- Add existing assets into Terraform.
- Use Terraform to create identical IT assets in a different region/geography.
- Given a scenario, test the recovery using the new infrastructure with high availability.

Lesson 4

High Availability & DR of Databases

- Explore log-shipping to a SQL DR instance.
- Use full geo-replication for SQL databases.
- Create automated backups for SQL databases.

Course 3

Self-Healing Architecture

Learn how to deploy microservices or cloud architecture that is resilient enough to withstand failures and predictable enough to resolve issues via automation without human intervention. This framework is known as self-healing architecture. Begin by learning some self-healing system design fundamentals such as single points of failure and three-tier architecture. Then we will show some self-healing deployment strategies, implementation steps, and use cases. Finally, we'll cover some cloud automation that learners can use to increase the resiliency of systems, such as auto-scaling automation.

Course Project

Deployment Roulette

Play the role of an engineer at a growing consulting firm. Applications left by a departing team are in an undocumented, unknown state. Identify failing applications and implement fixes to resolve the problems. Create an architecture diagram that communicates the status of the cloud environment to improve the onboarding of future developers.

Lesson 1

Design Self-Healing Systems & Visualize Them with Architecture Diagrams

- Identify single points of failure in system architecture and describe resolution strategies.
- Describe three-tier architecture benefits and drawbacks.
- Describe self-healing architecture automation strategies.
- Describe best-practice microservice design for self-healing architecture.
- Visualize self-healing system design by analyzing and creating diagrams.

Lesson 2

Implement Self-Healing Deployment Strategies

- Describe multiple deployment strategies and their benefits and drawbacks.
- Assess in which scenarios to use specific deployment strategies.
- Implement rolling, canary, and blue-green self-healing deployment strategies.

Lesson 3

Implement Scaling & Failover Automation Strategies for High-Availability Applications

- Describe cloud automation for scaling and failover.
- Automate microservices scaling.
- Automate virtual machines scaling.
- Automate microservice cluster scaling.

Course 4

Establishing a Culture of Reliability

This course is all about establishing a lasting culture focused on reliability. Learn how to develop processes and frameworks that will drive their workplace towards putting reliability first. Learners will begin by working through the incident management process and how to have effective on-calls. Following that, they will learn how to perform reliability reviews on various phases of a system. Next, they will learn how to effectively manage system capacity without being wasteful. We will round out this course with a lesson on how to reduce toil to free up time to focus on the work that matters.

Course Project

Plan, Reduce, Repeat

Participate in three mock scenarios one might encounter as an SRE. In the first scenario, utilize capacity management skills and demonstrate how to maintain an as-built document. In the second scenario, utilize on-call best practices and complete with a post-mortem. In the third scenario, develop a toil reduction plan and perform some hands-on automation.

Lesson 1

Improving On-Call Effectiveness

- Understand and utilize incident management process.
- Exhibit on-call best practices to have balanced and effective on-calls.
- Effectively write blameless post-mortems.

Lesson 2

Performing Reliability Reviews

- Explain how Zero Trust relates to infrastructure and networking.
- Design network resources to provide security borders.
- Configure Azure Bastion.
- Configure Just-in-time.
- Configure Azure Firewall.

Lesson 3

Managing System Capacity

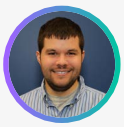
- Perform load test.
- Analyze capacity requirements.
- Utilize tiered capacity to effectively manage capacity for present, future, and emergency needs.
- Mitigate capacity risks by utilizing capacity management best practice.

Lesson 4

Toil Reduction

- Identify and measure toil.
- Employ common toil reduction strategies.
- Develop and execute a toil reduction plan.

Meet your instructors.



Nathan Anderson, MBA

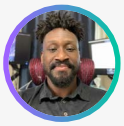
Global Cloud Architect

Nathan is a Certified Six Sigma Black Belt and has 10+ years of experience in IT in multiple industries. He is also the Instructor for two other Udacity courses: Ensuring Quality Releases and Azure Performance.

Travis Scotto

Site Reliability Engineer

Travis Scotto has worked in technology for 10 years. He has worked in various infrastructure roles: virtualization, databases, and monitoring. As an SRE, he employs automation and monitoring daily. He also has adjunct taught IT classes for 4.5 years.



Emmanuel Apau

CTO at Mechanicode.io

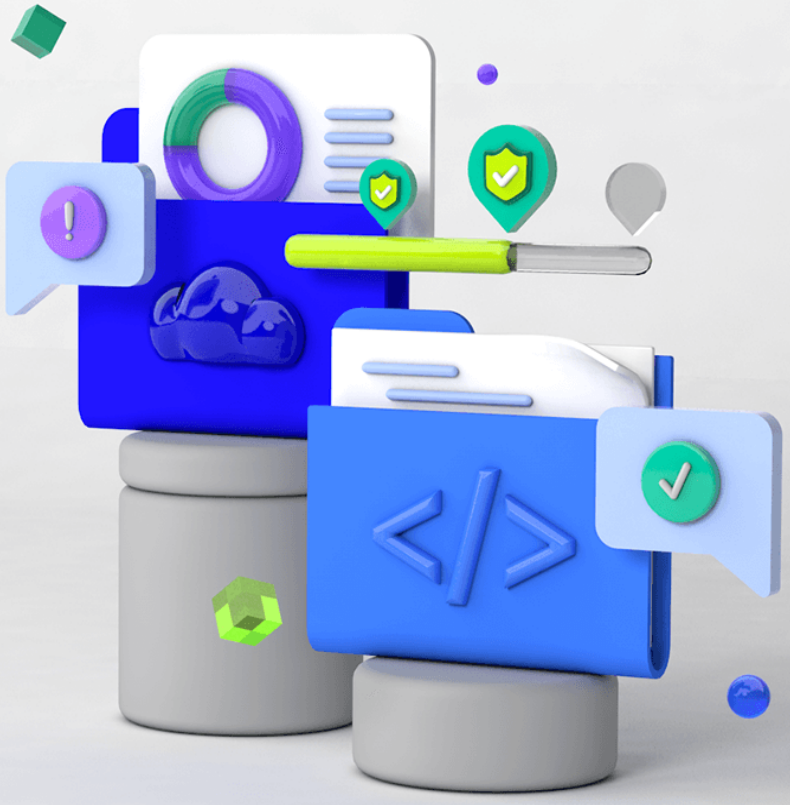
Emmanuel is cofounder of the Black Code Collective and DC's Technical.ly RealLIST Engineer award recipient. An AWS Certified DevSecOps specialist with 12 years of experience, he has spent his career developing innovative solutions using DevSecOps and site reliability best practices.

Sonny Sevin

Site Reliability Engineer

Sonny is an SRE with a varied background. He has dabbled in research at Lawrence Berkeley National Labs before moving into site reliability engineering to have a more hands on role. He has been published in several computing journals, as well as taught introductory programming courses.

Udacity's learning experience



Hands-on Projects

Open-ended, experiential projects are designed to reflect actual workplace challenges. They aren't just multiple choice questions or step-by-step guides, but instead require critical thinking.

Quizzes

Auto-graded quizzes strengthen comprehension. Learners can return to lessons at any time during the course to refresh concepts.

Knowledge

Find answers to your questions with Knowledge, our proprietary wiki. Search questions asked by other students, connect with technical mentors, and discover how to solve the challenges that you encounter.

Custom Study Plans

Create a personalized study plan that fits your individual needs. Utilize this plan to keep track of movement toward your overall goal.

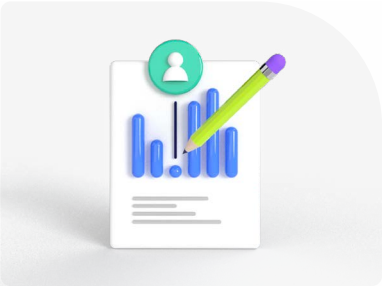
Workspaces

See your code in action. Check the output and quality of your code by running it on interactive workspaces that are integrated into the platform.

Progress Tracker

Take advantage of milestone reminders to stay on schedule and complete your program.

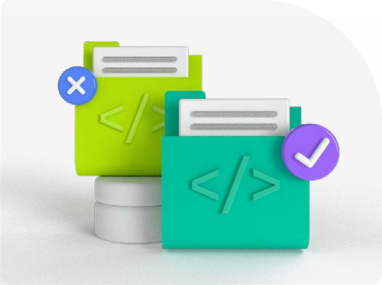
Our proven approach for building job-ready digital skills.



Pre-Assessments

Identify skills gaps.

- In-depth assessments benchmark your team's current level of knowledge in key areas.
- Results are used to generate custom learning paths.



Experienced Project Reviewers

Verify skills mastery.

- Personalized project feedback and critique includes line-by-line code review from skilled practitioners with an average turnaround time of 1.1 hours.
- Project review cycle creates a feedback loop with multiple opportunities for improvement—until the concept is mastered.
- Project reviewers leverage industry best practices and provide pro tips.

Technical Mentor Support

24/7 support unblocks learning.

- Learning accelerates as skilled mentors identify areas of achievement and potential for growth.
- Unlimited access to mentors means help arrives when it's needed most.
- 2 hr or less average question response time assures that skills development stays on track.

Mentor Network

Highly vetted for effectiveness.

- Mentors must complete a 5-step hiring process to join Udacity's selective network.
- After passing an objective and situational assessment, mentors must demonstrate communication and behavioral fit for a mentorship role.
- Mentors work across more than 30 different industries and often complete a Nanodegree program themselves.

Dashboard & Reporting

Track course progress.

- Udacity's enterprise management console simplifies management of bulk enrollments and employee onboarding.
- Interactive views help achieve targeted results to increase retention and productivity.
- Maximize ROI while optimizing job readiness.

UDACITY

Learn more at

udacity.com/enterprise →

01.13.22 | V1.0