Site Reliability Engineering (SRE) Certified Professional Training

About
DevOpsSchool

DevOpsSchool is a unit of "Cotocus PVT ltd" and a leading platform which helps IT organizations and professionals to learn all the emerging technologies and trend which helps them to learn and embrace all the skills, intelligence, innovation and transformation which requires to achieve the end result, quickly and efficiently. We provide over 40 specialized programs on DevOps, Cloud, Containers, Security, AI, ML and on Big data that are focused on industry requirement and each curriculum is developed and delivered by leading experts in each domain and aligned with the industry standards.

About Course

The Site Reliability Engineering Certified Professional (SRECP) certification course by DevOpsSchool will help you to learn the principles & practices that allows an organization to reliably and economically scale critical services. SRE is a process of operations which emphasize to accumulate software engineering and automation solutions to ensure that continuously delivered applications are running efficiently and reliably. Our SRECP course highlights the progression of SRE in modern software engineering process and its future direction and prepares learners with the methods, practices, and tools to engage workforce across the organization involved in reliability and stability evidenced through the use of real-life scenarios and case stories.

Hi, I am a Site Reliability Engineer (SRE)

"The Site Reliability Engineering Certified Professional (SRECP) certificationcourse by DevOpsSchool will help you to learn the principles & practices that allows an organization to reliably and economically scale critical services."

Co-ordinator – Akanksha

Call/WhatsApp: - +91 1800 889 7977

Mail Address: - contact@DevOpsSchool.com

Secondary contact – Patrick

Call/WhatsApp: - +91 7004 215 841

Mail Address:- contact@DevOpsSchool.com

Duration	72 Hrs
Mode	Online (Instructor-led, live & Interactive)
Projects (Real time scenario based)	1

FEATURES	DEVOPSSCHOOL	OTHERS
Faculty Profile Check	✔	✘
Lifetime Technical Support	✔	✘
Lifetime LMS access	✔	✘
Top 26 Tools	✔	✘
Training + Additional Videos	✔	✘
Real time scenario projects	✔	✘
Interview KIT (Q&A)	✔	✘
Training Notes	✔	✘
Step by Step Web Based Tutorials	✔	✘
Training Slides	✔	✘

Training

Upon completion of this program you will get 360-degree understanding of Site Reliability Engineering (SRE). This course will give you thorough learning experience in terms of understanding the concepts, mastering them thoroughly and applying them in real work environment.

Projects

You will be given industry level real time projects to work on and it will help you to differentiate yourself with multi-platform fluency, and have real-world experience with the most important tools and platforms.

Interview

As part of this, You would be given complete interview preparations kit, set to be ready for the Site Reliability Engineering (SRE) hotseat. This kit has been crafted by 200+ years industry experience and the experiences of nearly 10000 DevOpsSchool's DevOps learners worldwide.

Agenda/Course outline of the MDE Training (DevOps/DevSecOps/SRE)

SDLC Models & Architecture with Agile, DevOps, SRE & DevSecOps, SOA & Micro services - Concept

- Let's Understand about Software Development Model
- Overview of Waterfall Development Model
- Challenges of Waterfall Development Model
- Overview of Agile Development Model
- Challenges of Agile Development Model
- Requirement of New Software Development Model
- Understanding an existing Pain and Waste in Current Software Development Model
- What is DevOps?
- Transition in Software development model
  - Waterfall -> Agile -> CI/CD -> DevOps -> DevSecOps
- Understand DevOps values and principles
- Culture and organizational considerations
- Communication and collaboration practices
- Improve your effectiveness and productivity
- DevOps Automation practices and technology considerations
- DevOps Adoption considerations in an enterprise environment
- Challenges, risks and critical success factors
- What is DevSecOps?
  - Let's Understand DevSecOps Practices and Toolsets.
- What is SRE?
  - Let's Understand SRE Practices and Toolsets.
- List of Tools to become Full Stack Developer/QA/SRE/DevOps/DevSecOps
- Microservices Fundamentals
- Microservices Patterns
  - Choreographing Services
  - Presentation components
  - Business Logic

- - - o Database access logic

    o Application Integration

    o Modelling Microservices

    o Integrating multiple Microservices

    o Avoiding Breaking Changes

    o Choosing the right protocols

- Keeping it simple
    o Sync & Async

    o Dealing with legacy systems

    o Testing

- What and When to test

- Preparing for deployment

- Monitoring Microservice Performance

- Tools used for Microservices Demo using container

Platform - Operating Systems - Centos/Ubuntu & VirtualBox & Vagrant

- Installing CentOS7 and Ubuntu
- Accessing Servers with SSH
- Working at the Command Line
- Reading Files
- Using the vi Text Editor
- Piping and Redirection
- Archiving Files
- Accessing Command Line Help
- Understanding File Permissions
- Accessing the Root Account
- Using Screen and Script
- Overview of Hypervisor
- Introduction of VirtualBox
- Install VirtualBox and Creating CentOS7 and Ubuntu Vms
- Understanding Vagrant
- Basic Vagrant Workflow
- Advance Vagrant Workflow
- Working with Vagrant VMs
- The Vagrantfile
- Installing Nginx

- Provisioning
- Networking
- Sharing and Versioning Web Site Files
- Vagrant Share
- Vagrant Status
- Sharing and Versioning Nginx Config Files
- Configuring Synced Folders

Platform - Cloud - AWS - 4 hrs

- Introduction of AWS
- Understanding AWS infrastructure
- Understanding AWS Free Tier
- IAM: Understanding IAM Concepts
- IAM: A Walkthrough IAM
- IAM: Demo & Lab
- Computing:EC2: Understanding EC2 Concepts
- Computing:EC2: A Walkthrough EC2
- Computing:EC2: Demo & Lab
- Storage:EBS: Understanding EBS Concepts
- Storage:EBS: A Walkthrough EBS
- Storage:EBS: Demo & Lab
- Storage:S3: Understanding S3 Concepts
- Storage:S3: A Walkthrough S3
- Storage:S3: Demo & Lab
- Storage:EFS: Understanding EFS Concepts
- Storage:EFS: A Walkthrough EFS
- Storage:EFS: Demo & Lab
- Database:RDS: Understanding RDS MySql Concepts
- Database:RDS: A Walkthrough RDS MySql
- Database:RDS: Demo & Lab
- ELB: Elastic Load Balancer Concepts
- ELB: Elastic Load Balancer Implementation
- ELB: Elastic Load Balancer: Demo & Lab
- Networking:VPC: Understanding VPC Concepts
- Networking:VPC: Understanding VPC components
- Networking:VPC: Demo & Lab

Platform - Containers - Docker - 4 hrs

- What is Containerization?
- Why Containerization?
- How Docker is good fit for Containerization?
- How Docker works?
- Docker Architecture
- Docker Installations & Configurations
- Docker Components
- Docker Engine
- Docker Image
- Docker Containers
- Docker Registry
- Docker Basic Workflow
- Managing Docker Containers
- Creating our First Image
- Understading Docker Images
- Creating Images using Dockerfile
- Managing Docker Images
- Using Docker Hub registry
- Docker Networking
- Docker Volumes
- Deepdive into Docker Images
- Deepdive into Dockerfile
- Deepdive into Docker Containers
- Deepdive into Docker Networks
- Deepdive into Docker Volumes
- Deepdive into Docker Volume
- Deepdive into Docker CPU and RAM allocations
- Deepdive into Docker Config
- Docker Compose Overview
- Install & Configure Compose
- Understanding Docker Compose Workflow
- Understanding Docker Compose Services
- Writing Docker Compose Yaml file
- Using Docker Compose Commands
- Docker Compose with Java Stake
- Docker Compose with Rails Stake
- Docker Compose with PHP Stake
- Docker Compose with Nodejs Stak

Planning and Designing - Jira & Confluence (2 + 2 = 4 hrs)

JIRA
- o Overview of Jira
- o Use cases of Jira
- o Architecture of Jira
- o Installation and Configuraration of Jira in Linux
- o Installation and Configuraration of Jira in Windows
- o Jira Terminologies
- o Understanding Types of Jira Projects
- o Working with Projects
- o Working with Jira Issues
- o Adding Project Components and Versions
- o Use Subtasks to Better Manage and Structure Your Issues
- o Link Issues to Other Resources
- o Working in an Agile project
- o Working with Issues Types by Adding/Editing/Deleting
- o Working with Custom Fields by Adding/Editing/Deleting
- o Working with Screens by Adding/Editing/Deleting
- o Searching and Filtering Issues
- o Working with Workflow basic
- o Introduction of Jira Plugins and Addons.
- o Jira Integration with Github

Confluence
- o Exploring Confluence benefits and resources
- o Configuring Confluence
- o Navigating the dashboard, spaces, and pages
- o Creating users and groups
- o Creating pages from templates and blueprints
- o Importing, updating, and removing content
- o Giving content feedback
- o Watching pages, spaces, and blogs
- o Managing tasks and notifications
- o Backing up and restoring a site
- o Admin tasks
  - ▪ Add/Edit/Delete new users
  - ▪ Adding group and setting permissions
  - ▪ Managing user permissions
  - ▪ Managing addons or plugins
  - ▪ Customizing confluence site

- o Installing Confluence
  - ▪ Evaluation options for Confluence
  - ▪ Supported platforms
  - ▪ Installing Confluence on Windows
  - ▪ Activating Confluence trial license
  - ▪ Finalizing Confluence Installation

Backend Programming Language 2 - Python/Flask with MySQL DB - 4 hrs.

- Planning - Discuss some of the Small Project Requirement which include Login/Registration with Some Students records CRUD operations.

- Design a Method --> Classes -> Interface using Core Python
  - o Fundamental of Core Python with Hello-world Program with Method -->Classes

- Coding in Flask using HTML - CSS - JS - MySQL
  - o Fundamental of Flask Tutorial of Hello-World APp

- UT - 2 Sample unit Testing using Python test

- Package a Python App

- AT - 2 Sample unit testing using Selenium

Deploy to Some env.
-----------------END OF THE CLASS ------------------------
Technology Demonstration

- o Software Planning and Designing using JAVA
- o Core Python
- o Flask
- o mySql
- o pytest
- o Selenium
- o HTML
- o CSS
- o Js.

Source Code Versioning - Git using Github - 4 hrs

- Introduction of Git

- Installing Git

- Configuring Git

- Git Concepts and Architecture

- How Git works?

- The Git workflow

- Working with Files in Git

  o Adding files

  o Editing files

  o Viewing changes with diff

  o Viewing only staged changes

  o Deleting files

  o Moving and renaming files

  o Making Changes to Files

- Undoing Changes

  o Reset

  o Revert

- Amending commits

- Ignoring Files

- Branching and Merging using Git

- Working with Conflict Resolution

- Comparing commits, branches and workspace

- Working with Remote Git repo using Github

- Push - Pull - Fetch using Github

- Tagging with Git

Code Analysis & Securing Code (SAST) - SonarQube & Coverity Scan & Snyk

- What is SonarQube?

- Benefits of SonarQube?

- Alternative of SonarQube

- Understanding Various License of SonarQube

- Architecture of SonarQube

- How SonarQube works?

- Components of SonarQube

- SonarQube runtime requirements

- Installing and configuring SonarQube in Linux

- Basic Workflow in SonarQube using Command line

- Working with Issues in SonarQube

- Working with Rules in SonarQube

- Working with Quality Profiles in SonarQube

- Working with Quality Gates in SonarQube

- Deep Dive into SonarQube Dashboad

- Understanding Seven Axis of SonarQube Quality

- Workflow in SonarQube with Maven Project

- Workflow in SonarQube with Gradle Project

- OWASP Top 10 with SonarQube

Build Management - Maven and Gradle - 2 + 2 = 4 hrs

Maven

- Introduction to Apache Maven
- Advantage of Apache Maven over other build tools
- Understanding the Maven Lifecycle and Phase
- Understanding the Maven Goals
- Understanding the Maven Plugins
- Understanding the Maven Repository
- Understanding and Maven Release and Version
- Prerequisite and Installing Apache Maven
- Understanding and using Maven Archetypes
- Understanding Pom.xml and Setting.xml
- Playing with multiples Maven Goals
- Introducing Maven Dependencies
- Introducing Maven Properties
- Introducing Maven Modules
- Introducing Maven Profile
- Introducing Maven Plugins
- How can Maven benefit my development process?
- How do I setup Maven?
- How do I make my first Maven project?
- How do I compile my application sources?
- How do I compile my test sources and run my unit tests?
- How do I create a JAR and install it in my local repository?
- How do I use plugins?
- How do I add resources to my JAR?
- How do I filter resource files?
- How do I use external dependencies?
- How do I deploy my jar in my remote repository?
- How do I create documentation?
- How do I build other types of projects?
- How do I build more than one project at once?

Gradle

- What is Gradle?
- Why Gradle?
- Installing and Configuring Gradle
- Build Java Project with Gradle
- Build C++ Project with Gradle
- Build Python Project with Gradle
- Dependency Management in Gradle
- Project Structure in Gradle
- Gradle Tasks
- Gradle Profile and Cloud
- Gradle Properties
- Gradle Plugins

Package Management - Packer & Artifactory - 2 + 2 = 4hrs

Artifactory

- Artifactory Overview
- Understanding a role of Artifactory in DevOps
- System Requirements
- Installing Artifactory in Linux
- Using Artifactory
- Getting Started
- General Information
- Artifactory Terminology
- Artifactory Repository Types
- Artifactory Authentication
- Deploying Artifacts using Maven
- Download Artifacts using Maven
- Browsing Artifactory
- Viewing Packages
- Searching for Artifacts
- Manipulating Artifacts

Packer

- Getting to Know Packer
  - o What is Packer?
  - o Save What is Packer?
  - o Installing Packer
  - o Save Installing Packer

- The Packer workflow and components
- Save The Packer workflow and components
- The Packer CLI
- Save The Packer CLI

- Baking a Website Image for EC2
    - Select an AWS AMI base
    - Save Select an AWS AMI base
    - Automate AWS AMI base build
    - Save Automate AWS AMI base build
    - Using build variables
    - Save Using build variables
    - Provision Hello World
    - Save Provision Hello World
    - Provision a basic site
    - Save Provision a basic site

- Customization with a Config Management Tool
    - Simplify provisioning with a config tool
    - Save Simplify provisioning with a config tool
    - Use ansible to install the webserver
    - Save Use ansible to install the webserver
    - Debugging
    - Save Debugging

- Building Hardened Images
    - Use Ansible modules to harden our image
    - Save Use Ansible modules to harden our image
    - Baking a Jenkins image
    - Save Baking a Jenkins image

- Building a Pipeline for Packer Image
    - Validate Packer templates
    - Save Validate Packer templates
    - Create a manifest profile
    - Save Create a manifest profile
    - Testing
    - Save Testing
    - CI pipeline
    - Save CI pipeline

Unit Testing & Acceptance Testing & Coverage - Junit & Selenium & Jmeter & Jacoco - 4hrs

- Junit Fundamental - 1 Hour
  - o What is Unit Testing
  - o Tools for Unit Testing
  - o What is Junit?
  - o How to configure Junit?
  - o Writing Basic Junit Test cases
  - o Running Basic Junit Test cases
  - o Junit Test Results

- Selenium Fundamental -2 Hours
  - o Introduction To Selenium
  - o Components of Selenium
    - Selenium IDE
    - Selenium Webdriver
    - Selenium Grid
  - o Installing and Configuring Selenium
  - o Working with Selenium IDE
  - o Working With Selenium Webdriver with Java Test Case
  - o Setup and Working with Selenium Grid

- Jacoco - 1 Hours
  - o Overview of Code Coavegar Process
  - o Introduction of Jacoco
  - o How Jacoco works!
  - o How to install Jaoco?
  - o Setup testing Envrionement with Jacoco
  - o Create test data files using Jacoco and Maven
  - o Create a Report using Jacoco
  - o Demo - Complete workflow of Jacoco with Maven and Java Project

Configuration & Deployment Management - Ansible - 4 hrs

- Overflow of Configuration Management
- Introduction of Ansible
- Ansible Architecture
- Let's get started with Ansible
- Ansible Authentication & Authorization
- Let's start with Ansible Adhoc commands
- Let's write Ansible Inventory
- Let's write Ansible Playbook
- Working with Popular Modules in Ansible
- Deep Dive into Ansible Playbooks
- Working with Ansible Variables
- Working with Ansible Template
- Working with Ansible Handlers
- Roles in Ansible
- Ansible Galaxy

Container Orchestration - Kubernetes & Helm Introduction - 4 hrs

Kubernetes

- Understanding the Need of Kubernetes
- Understanding Kubernetes Architecture
- Understanding Kubernetes Concepts
- Kubernetes and Microservices
- Understanding Kubernetes Masters and its Component
  - o kube-apiserver
  - o etcd
  - o kube-scheduler
  - o kube-controller-manager
- Understanding Kubernetes Nodes and its Component
  - o kubelet
  - o kube-proxy
  - o Container Runtime

- Understanding Kubernetes Addons
  - DNS
  - Web UI (Dashboard)
  - Container Resource Monitoring
  - Cluster-level Logging
- Understand Kubernetes Terminology
- Kubernetes Pod Overview
- Kubernetes ReplicationContrller Overview
- Kubernetes Deployment Overview
- Kubernetes Servcie Overview
- Understanding Kubernetes running environment options
- Working with first Pods
- Working with first ReplicationContrller
- Working with first Deployment
- Working with first Services
- Introducing Helm
- Basic working with Helm

Infrastructure Coding - Terraform - 4 hrs

- Deploying Your First Terraform Configuration
  - Introduction
  - What's the Scenario?
  - Terraform Components
- Updating Your Configuration with More Resources
  - Introduction
  - Terraform State and Update
  - What's the Scenario?
  - Data Type and Security Groups
- Configuring Resources After Creation
  - Introduction
  - What's the Scenario?
  - Terraform Provisioners
  - Terraform Syntax

- Adding a New Provider to Your Configuration
  - o Introduction
  - o What's the Scenario?
  - o Terraform Providers
  - o Terraform Functions
  - o Intro and Variable
  - o Resource Creation
  - o Deployment and Terraform Console
  - o Updated Deployment and Terraform Commands

Continuous Integration - Jenkins - 4 hrs.

- Let's understand Continuous Integration
  - o What is Continuous Integration
  - o Benefits of Continuous Integration
  - o What is Continuous Delivery
  - o What is Continuous Deployment
  - o Continuous Integration Tools
- What is Jenkins
  - o History of Jenkins
  - o Jenkins Architecture
  - o Jenkins Vs Jenkins Enterprise
  - o Jenkins Installation and Configurations
- Jenkins Dashboard Tour
  - o Understand Freestyle Project
  - o Freestyle General Tab
  - o Freestyle Source Code Management Tab
  - o Freestyle Build Triggers Tab
  - o Freestyle Build Environment
  - o Freestyle Build
  - o Freestyle Post-build Actions
  - o Manage Jenkins
  - o My Views & Credentials
  - o People & Build History

- Creating a Simple Job
  - Simple Java and Maven Based Application
  - Simple Java and Gradle Based Application
  - Simple DOTNET and MSBuild Based Application
- Jobs Scheduling in Jenkins
  - Manually Building
  - Build Trigger based on fixed schedule
  - Build Trigger by script
  - Build Trigger Based on pushed to git
- Useful Jobs Configuration
  - Jenkins Jobs parameterised
  - Execute concurrent builds
  - Jobs Executors
  - Build Other Projects
  - Build after other projects are built
  - Throttle Builds
- Jenkins Plugins
  - Installing a Plugin
  - Plugin Configuration
  - Updating a Plugin
  - Plugin Wiki
  - Top 20 Useful Jenkins Plugins
  - Using Jenkins Pluginss Best Practices
- Jenkins Node Management
  - Adding a Linux Node
  - Adding a Windows Nodes
  - Nodes Management using Jenkins & Jenkins Nodes High Availability
- Jenkins Integration with other tools
  - Jira
  - Git
  - SonarQube
  - Maven
  - Junit
  - Ansible

- o Docker
- o AWS
- o Jacoco
- o Coverity
- o Selenium
- o Gradle
- **Reports in Jenkins**
  - o Junit Report
  - o SonarQube Reports
  - o Jacoco Reports
  - o Coverity Reports
  - o Selenium Reports
  - o Test Results
  - o Cucumber Reports
- **Jenkins Node Management**
  - o Adding a Linux Node
  - o Adding a Windows Nodes
  - o Nodes Management using Jenkins
  - o Jenkins Nodes High Availability
- **Notification & Feedback in Jenkins**
  - o CI Build Pipeline & Dashboard
  - o Email Notification
  - o Advance Email Notification
  - o Slack Notification
- **Jenkins Advance – Administrator**
  - o Security in Jenkins
  - o Authorization in Jenkins
  - o Authentication in Jenkins
  - o Managing folder/subfolder
  - o Jenkins Upgrade
  - o Jenkins Backup
  - o Jenkins Restore
  - o Jenkins Command Line

Infrastructure Monitoring Tool 2 - Datadog - 4 hrs.

Datadog – 3 hrs.

- Introduction
  - o Introduction to Datadog
  - o Datadog installation
  - o Grafana with Datadog Installation
- Monitoring
  - o Introduction to Monitoring
  - o Client Libraries
  - o Pushing Metrics
  - o Querying
  - o Service Discovery
  - o Exporters
- Alerting
  - o Introduction to Alerting
  - o Setting up Alerts
- Internals
  - o Datadog Storage
  - o Datadog Security
  - o TLS & Authentication on Datadog Server
  - o Mutual TLS for Datadog Targets
- Use Cases
  - o Monitoring a web application
  - o Calculating Apdex score
  - o Cloudwatch Exporter
  - o Grafana Provisioning
  - o Consul Integration with Datadog
  - o EC2 Auto Discovery
  - o Installing on Mac
  - o Installing using Docker
  - o Building from source
  - o Upgrading
- Administration

- o Configuration
- o Authentication
- o Permissions
- o Grafana CLI
- o Internal metrics
- o Provisioning & Troubleshooting

Log Monitoring Tool 1 - Splunk - 4 hrs

- What Is Splunk?
  - o Overview
  - o Machine Data
  - o Splunk Architecture
  - o Careers in Splunk
- Setting up the Splunk Environment
  - o Overview
  - o Splunk Licensing
  - o Getting Splunk
  - o Installing Splunk
  - o Adding Data to Splunk
- Basic Searching Techniques
  - o Adding More Data
  - o Search in Splunk
  - o Demo: Splunk Search
  - o Splunk Search Commands
  - o Splunk Processing Language
  - o Splunk Reports
  - o Reporting in Splunk
  - o Splunk Alerts
  - o Alerts in Splunk

- Enterprise Splunk Architecture
  - Overview
  - Forwarders
  - Enterprise Splunk Architecture
  - Installing Forwarders
  - Installing Forwarders
  - Troubleshooting Forwarder Installation
- Splunking for DevOps and Security
  - Splunk in DevOps
  - DevOps Demo
  - Splunk in Security & Enterprise Use Cases
- Application Development in Splunkbase
  - What Is Splunkbase?
  - Navigating the Splunkbase
  - Creating Apps for Splunk
  - Benefits of Building in Splunkbase
- Splunking on Hadoop with Hunk
  - What Is Hadoop?
  - Running HDFS Commands
  - What Is Hunk?
  - Installing Hunk
  - Moving Data from HDFS to Hunk
  - Composing Advanced Searches
    - Splunk Searching
    - Introduction to Advanced Searching
    - Eval and Fillnull Commands
    - Other Splunk Command Usage
    - Filter Those Results! & The Search Job Inspector
  - Creating Search Macros
    - What Are Search Macros?
    - Using Search Macros within Splunk
    - Macro Command Options and Arguments
    - Other Advanced Searching within Splunk

Performance & RUM Monitoring - NewRelic - 4 hrs

- Introduction and Overview of NewRelic

  - What is Application Performance Management?
  - Understanding a need of APM
  - Understanding transaction traces
  - What is Application Performance?
  - APM Benefits
  - APM Selection Criteria
  - Why NewRelic is best for APM?
  - What is NewRelic APM? & How does NewRelic APM work?
  - NewRelic Architecture & Terminology

- Installing and Configuring NewRelic APM Agents for Application

  - Register a Newrelic Trial account
  - Installing a JAVA Agent to Monitor your Java Application
  - Installing a PHP Agent to Monitor your PHP Application
  - Installing New Relic Agent for .NET Framework Application
  - Installing a Docker based Agent to Monitor your Docker based Application
  - Understanding of NewRelic Configuration settings of newrelic.yml
  - Understanding of NewRelic Agent Configuration settings

- Working with NewRelic Dashboard

  - Understanding a transactions
  - Understanding Apdex and Calculating and Setting Apdex Threshold
  - Understanding Circuit break
  - Understanding Throughput
  - Newrelic default graphs
  - Understanding and Configuring Service Maps
  - Understanding and Configuring JVM
  - Understanding Error Analytics
  - Understanding Violations
  - Understanding and Configuring Deployments
  - Understanding and Configuring Thread Profiler
  - Depp Dive into Transaction Traces

- o   Profiling with New Relic
- o   Creating and managing Alerts
- o   Working with Incidents
- o   Sending NewRelic Alerts to Slack
- o   Assessing the quality of application deployments
- Monitoring using Newrelic
    - o   View your applications index
    - o   APM Overview page
    - o   New Relic APM data in Infrastructure
    - o   Transactions page
    - o   Databases and slow queries
    - o   Viewing slow query details & External services page
    - o   Agent-specific UI & Viewing the transaction map
- Deep Dive into Newrelic Advance
    - o   Newrelic transaction alerts
    - o   Configure abnd Troubleshoot and Cross Application Traces
    - o   NewRelic Service Level Agreements
    - o   Troubleshooting NewRelic
    - o   Understanding and Configuring NewRelic X-Ray Sessions
    - o   Deep Dive into NewRelic Agent Configuration
    - o   Adding Custom Data with the APM Agent
    - o   Extending Newrelic using Plugins
    - o   Finding and Fixing Application Performance Issues with New Relic APM
    - o   Setting up database montioring using Newrelic APM
    - o   Setting up and Configuring Newrelic Alerts
- Working with NewRelic Performance Reports
    - o   Availability report
    - o   Background jobs analysis report
    - o   Capacity analysis report
    - o   Database analysis report
    - o   Host usage report
    - o   Scalability analysis report
    - o   Web transactions analysis report
    - o   Weekly performance report

Webserver - Apache HTTP & Nginx

Apache HTTP

- Introduction to web server
- Install Apache on CentOS 7.4
- Enable Apache to automatically start when system boot
- Configure the firewall service
- Where is Apache?
- Directory structure
  - o Apache directory structure
  - o Configuration file
  - o Create your first page
- Virtual hosts
  - o Setting up the virtual host - name based
  - o Setting up the virtual host - port based
- Using aliases and redirecting
- Configuring an alias for a url
- Redirects
- Logging
  - o The error log
  - o The access log
  - o Custom log
  - o Log rotation
- Security
  - o Basic Security - Part 1
  - o Basic Security - Part 2
  - o Set up TLS/SSl for free
  - o Basic authentication
  - o Digest authentication
  - o Access Control
  - o .htaccess (Administrator Side)
  - o .htaccess (User Side)
  - o Install and Configure antivirus
  - o Mitigate dos attacks - mod_evasive
- Apache Performance and Troubleshooting
  - o Apache Multi-Processing Modules (MPMs)
  - o Adjusting httpd.conf - Part 1
  - o Adjusting httpd.conf - Part 2
  - o Troubleshoot Apache (Analyz Access Log) - Part 1
  - o Troubleshoot Apache (Analyze Access Log) - Part 2
  - o Use Apachetop to monitor web server traffic

Nginx

- Overview

- o Introduction
- o About NGINX
- o NGINX vs Apache
- o Test your knowledge
- Installation
  - o Server Overview
  - o Installing with a Package Manager
  - o Building Nginx from Source & Adding Modules
  - o Adding an NGINX Service
  - o Nginx for Windows
  - o Test your knowledge
- Configuration
  - o Understanding Configuration Terms
  - o Creating a Virtual Host
  - o Location blocks
  - o Variables
  - o Rewrites & Redirects
  - o Try Files & Named Locations
  - o Logging
  - o Inheritance & Directive types
  - o PHP Processing
  - o Worker Processes
  - o Buffers & Timeouts
  - o Adding Dynamic Modules
  - o Test your knowledge
- Performance
  - o Headers & Expires
  - o Compressed Responses with gzip
  - o FastCGI Cache
  - o HTTP2
  - o Server Push
- Security
  - o HTTPS (SSL)
  - o Rate Limiting
  - o Basic Auth
  - o Hardening Nginx
  - o Test your knowledge
  - o Let's Encrypt - SSL Certificates

Multi-cluster Kubernetes orchestration platform - Rancher

- Multi-cluster management
  - o Rancher provides a unified interface for managing multiple Kubernetes clusters across different environments, including on-premises, cloud, and hybrid.
- Centralized administration

- - With Rancher, you can manage user access, security policies, and cluster settings from a central location, making it easier to maintain a consistent and secure deployment across all clusters.
- Automated deployment
    - Rancher streamlines the application deployment process by providing built-in automation tools that allow you to deploy applications to multiple clusters with just a few clicks.
- Monitoring and logging
    - Rancher provides a built-in monitoring and logging system that enables you to monitor the health and performance of your applications and clusters in real-time.
- Application catalog
    - Rancher offers a curated catalog of pre-configured application templates that enable you to deploy and manage popular applications such as databases, web servers, and messaging queues.
- Scalability and resilience
    - Rancher is designed to be highly scalable and resilient, enabling you to easily add new clusters or nodes to your deployment as your needs grow.
- Extensibility
    - Rancher provides an open API and a rich ecosystem of plugins and extensions, enabling you to customize and extend the platform to meet your specific needs.

Services mesh Data planes & Control Planes - Envoy & Istio

Envoy

Data Plane

- Envoy is a high-performance proxy that is deployed as a sidecar to each microservice in the infrastructure.
- Envoy manages all inbound and outbound traffic for the microservice and provides features like load balancing, circuit breaking, and health checks.
- Envoy can also be used as a standalone proxy outside of a service mesh architecture.

Control Plane:

- Envoy does not have a built-in control plane.
- It can be integrated with other service mesh management solutions like Istio, Consul, or Linkerd, which provide a central point of management for the Envoy proxies.
- These control planes enable features like traffic management, security, and observability.

Istio:

Data Plane:

- Istio uses Envoy as its data plane, which means that each microservice has an Envoy sidecar proxy that manages the inbound and outbound traffic for that service.
- Envoy is configured and managed by Istio's control plane components.

Control Plane:

- Istio provides a built-in control plane that includes the following components:
- Pilot: responsible for managing the configuration of the Envoy proxies and enabling features like traffic routing and load balancing.
- Mixer: provides policy enforcement, telemetry collection, and access control for the microservices in the service mesh.
- Citadel: responsible for managing the security of the service mesh, including mutual TLS encryption and identity-based access control.

Network configurations and Service Discovery - Consul

Network Configurations:

- Consul provides a central service registry that keeps track of all the services in the infrastructure.

- Each microservice in the infrastructure registers itself with Consul, providing information like its IP address, port, and health status.

- Consul also supports multiple datacenters, allowing for the deployment of services across different regions or availability zones.

- Consul provides a DNS interface that can be used to discover services in the infrastructure. Applications can use this interface to resolve service names to IP addresses and connect to the appropriate service.

Service Discovery:

- Consul provides a service discovery mechanism that enables microservices to discover and communicate with each other.

- Consul supports different service discovery methods, including DNS, HTTP, and gRPC.

- Consul can perform health checks on the services in the infrastructure to ensure that they are functioning properly. If a service fails a health check, it is removed from the service registry until it is healthy again.

- Consul also supports service segmentation, allowing services to be grouped into logical subsets based on tags or other attributes. This enables more fine-grained control over service discovery and traffic routing.

Securing Credentials - HashiCorp Vault

Secret Storage:

- Vault provides a secure storage mechanism for sensitive data, including credentials, API keys, and other secrets.
- Vault uses encryption and access control policies to ensure that secrets are protected both at rest and in transit.
- Vault supports different storage backends, including disk, cloud storage, and key management systems.

Authentication:

- Vault provides several authentication methods that can be used to validate user or machine identity.
- These methods include LDAP, Active Directory, Kubernetes, and token-based authentication.
- Vault also supports multi-factor authentication (MFA) to provide an additional layer of security.

Access Control:

- Vault provides fine-grained access control policies that can be used to restrict access to specific secrets or resources.
- These policies can be based on user or machine identity, time of day, and other factors.
- Vault supports role-based access control (RBAC) and attribute-based access control (ABAC) policies.

Encryption:

- Vault provides end-to-end encryption for all secrets stored in its storage backend.
- Vault uses encryption keys that are stored separately from the secrets themselves, providing an additional layer of security.
- Vault supports different encryption algorithms and key management systems.

Auditing and Logging:

- Vault provides detailed auditing and logging capabilities that can be used to track access to secrets and detect potential security threats.
- Vault logs all user and system activity, including authentication events, secret access, and configuration changes.
- Vault also supports integration with popular logging and monitoring tools.

Infrastructure Monitoring Tool - Prometheus with Grafana

Prometheus

- Introduction
- Introduction to Prometheus
- Prometheus installation
- Grafana with Prometheus Installation

- Monitoring
- Introduction to Monitoring
- Client Libraries
- Pushing Metrics
- Querying
- Service Discovery
- Exporters

- Alerting
- Introduction to Alerting
- Setting up Alerts

- Internals
- Prometheus Storage
- Prometheus Security
- TLS & Authentication on Prometheus Server
- Mutual TLS for Prometheus Targets

- Use Cases
- Monitoring a web application
- Calculating Apdex score
- Cloudwatch Exporter
- Grafana Provisioning
- Consul Integration with Prometheus
- EC2 Auto Discovery

Grafana

- Installation
- Installing on Ubuntu / Debian
- Installing on Centos / Redhat
- Installing on Windows
- Installing on Mac
- Installing using Docker
- Building from source
- Upgrading

- Administration
- Configuration
- Authentication
- Permissions
- Grafana CLI
- Internal metrics
- Provisioning
- Troubleshooting

Log Monitoring Tool - Elasticsearch Logstash Kibana(ELK stake)

- Introduction to Elasticsearch
- Overview of the Elastic Stack (ELK+)
- Elastic Stack

- Architecture of Elasticsearch
- Nodes & Clusters
- Indices & Documents
- A word on types
- Another word on types
- Sharding
- Replication
- Keeping replicas synchronized

- Searching for data

- Distributing documents across shards

- Installing Elasticsearch & Kibana

- Running Elasticsearch & Kibana in Elastic Cloud

- Installing Elasticsearch on Mac/Linux

- Using the MSI installer on Windows

- Installing Elasticsearch on Windows

- Configuring Elasticsearch

- Installing Kibana on Mac/Linux

- Installing Kibana on Windows

- Configuring Kibana

- Kibana now requires data to be available

- Introduction to Kibana and dev tools

- Managing Documents
- Creating an index
- Adding documents

- Retrieving documents by ID

- Replacing documents

- Updating documents

- Scripted updates

- Upserts

- Deleting documents

- Deleting indices

- Batch processing

- Importing test data with cURL

- Exploring the cluster

- Mapping

- Introduction to mapping

- Dynamic mapping

- Meta fields

- Field data types

- Adding mappings to existing indices

- Changing existing mappings

- Mapping parameters

- Adding multi-fields mappings

- Defining custom date formats

- Picking up new fields without dynamic mapping

- Analysis & Analyzers

- Introduction to the analysis process

- A closer look at analyzers

- Using the Analyze API

- Understanding the inverted index

- Analyzers

- Overview of character filters

- Overview of tokenizers

- Overview of token filters

- Overview of built-in analyzers

- Configuring built-in analyzers and token filters

- Creating custom analyzers

- Using analyzers in mappings

- Adding analyzers to existing indices

- A word on stop words

- Introduction to Searching

- Search methods

- Searching with the request URI

- Introducing the Query DSL

- Understanding query results

- Understanding relevance scores

- Debugging unexpected search results

- Query contexts

- Full text queries vs term level queries

- Basics of searching

- Term Level Queries

- Introduction to term level queries

- Searching for a term

- Searching for multiple terms

- Retrieving documents based on IDs

- Matching documents with range values

- Working with relative dates (date math)

- Matching documents with non-null values

- Matching based on prefixes

- Searching with wildcards

- Searching with regular expressions

- Term Level Queries

- Full Text Queries

- Introduction to full text queries

- Flexible matching with the match query

- Matching phrases

- Searching multiple fields

- Full Text Queries

- Adding Boolean Logic to Queries

- Introduction to compound queries

- Querying with boolean logic

- Debugging bool queries with named queries

- How the "match" query works

Incident Response Tool - PagerDuty & Opsgenie

Alert Management:

- Both PagerDuty and Opsgenie provide powerful alert management capabilities, allowing teams to configure alerts based on specific criteria, such as event severity, priority, and more.

- Alerts can be sent to multiple channels, including email, SMS, voice, and mobile push notifications.

- Both tools also provide support for escalation policies, allowing teams to ensure that critical alerts are addressed promptly.

Incident Management:

- Both tools also provide support for escalation policies, allowing teams to ensure that critical alerts are addressed promptly.

- Both PagerDuty and Opsgenie provide incident management capabilities, allowing teams to track incidents and collaborate on resolving them.

- Incident management features include creating incidents, adding notes, assigning owners, and tracking status changes.

- Both tools also provide support for incident timelines, allowing teams to visualize the progress of an incident over time.

Integration:

- Both PagerDuty and Opsgenie provide extensive integration capabilities, allowing teams to integrate with a wide range of tools and technologies.

- Integrations include popular monitoring tools, such as Nagios, New Relic, and AWS CloudWatch, as well as IT service management (ITSM) tools like JIRA and ServiceNow.

- Both tools also provide REST APIs for custom integrations.

Analytics and Reporting:

- Both PagerDuty and Opsgenie provide analytics and reporting capabilities, allowing teams to track performance metrics and identify areas for improvement.

- Analytics and reporting features include incident duration, resolution times, and other key performance indicators (KPIs).

- Both tools also provide support for custom dashboards and reports.

Automation:

- Both PagerDuty and Opsgenie provide automation capabilities, allowing teams to automate repetitive tasks and streamline incident response processes.

- Automation features include auto-acknowledgment of alerts, auto-escalation of incidents, and auto-remediation of issues.

- Both tools also provide support for scripting and custom automation workflows.

Production Env Job scheduler and Run Book Automation - RunDeck

Job Scheduling:

- RunDeck provides powerful job scheduling capabilities, allowing teams to schedule jobs based on specific criteria, such as time, date, and recurrence.
- Jobs can be executed on multiple platforms, including Windows, Linux, and macOS.
- RunDeck also provides support for job dependencies, allowing teams to ensure that jobs are executed in the correct order.

Run Book Automation:

- RunDeck provides run book automation capabilities, allowing teams to automate repetitive tasks and streamline operations.
- Run book automation features include executing commands, scripts, and workflows on multiple systems, as well as orchestrating complex processes across multiple systems.
- RunDeck also provides support for auditing and logging, allowing teams to track changes and monitor system activity.

Integration:

- RunDeck provides extensive integration capabilities, allowing teams to integrate with a wide range of tools and technologies.
- Integrations include popular configuration management tools, such as Ansible and Puppet, as well as monitoring tools like Nagios and Zabbix.
- RunDeck also provides REST APIs for custom integrations.

Access Control:

- RunDeck provides access control capabilities, allowing teams to control who can access and execute jobs and workflows.
- Access control features include role-based access control (RBAC), LDAP integration, and multi-factor authentication (MFA).
- RunDeck also provides support for audit logging, allowing teams to track user activity and changes to system configurations.

Notifications and Reporting:

- RunDeck provides notifications and reporting capabilities, allowing teams to track performance metrics and identify areas for improvement.

- Notifications and reporting features include job execution status, error notifications, and custom reports.
- RunDeck also provides support for custom dashboards and reports.

Application Performance Monitoring - Appdynamics

Application Performance Monitoring:

- AppDynamics provides powerful application performance monitoring capabilities, allowing teams to monitor the performance of their applications in real-time.
- APM features include application topology maps, transaction tracing, code-level diagnostics, and performance baselines.
- AppDynamics also provides support for identifying and troubleshooting performance issues, such as slow database queries, inefficient code, and memory leaks.

End-User Monitoring:

- AppDynamics provides end-user monitoring capabilities, allowing teams to track the performance of their applications from the end-user perspective.
- End-user monitoring features include real-user monitoring, synthetic monitoring, and business transaction monitoring.
- AppDynamics also provides support for identifying and troubleshooting end-user issues, such as slow page load times and errors.

Infrastructure Monitoring:

- AppDynamics provides infrastructure monitoring capabilities, allowing teams to monitor the health and performance of their infrastructure.
- Infrastructure monitoring features include server monitoring, container monitoring, and cloud monitoring.
- AppDynamics also provides support for identifying and troubleshooting infrastructure issues, such as high CPU usage, low memory, and network latency.

Integration:

- AppDynamics provides extensive integration capabilities, allowing teams to integrate with a wide range of tools and technologies.
- Integrations include popular monitoring tools, such as Splunk and Elasticsearch, as well as IT service management (ITSM) tools like ServiceNow and Remedy.
- AppDynamics also provides REST APIs for custom integrations.

Analytics and Reporting:

- AppDynamics provides analytics and reporting capabilities, allowing teams to track performance metrics and identify areas for improvement.

- Analytics and reporting features include transaction analysis, error analysis, and custom dashboards.

- AppDynamics also provides support for machine learning and predictive analytics, allowing teams to proactively identify and address performance issues.

Thank you!

Connect with us for more info

Call/WhatsApp: - +91 700 483 5930
Mail: - contact@DevOpsSchool.com
www.DevOpsSchool.com