Cloud Native Computing Specialist

A ONE YEAR CLOUD COMPUTING PROGRAM DESIGNED FORABSOLUTE BEGINNERS.
GETTING PAKISTAN READY FOR THE NEW ERA OF MICROSERVICES AND MULTI-CLOUD NATIVE COMPUTING.

Program Structure

A four-quarter cloud computing program.

Quarter 1
Cloud Native Computing Foundations

Quarter 2
Developing Cloud Native Microservices

Quarter 3
Kubernetes in Practice

Quarter 4
AWS Application Development

Detailed Program Structure

1A

Introduction to Cloud Native Computing

Cloud native computing uses open-source software stacks to deploy applications as microservices, packaging each part into its own container and dynamically orchestrating those containers to optimize resource utilization. In the last 15 years, cloud application delivery has moved from being bound to physical servers to running on virtual machines with a full operating system and now to containers with Docker, where developers can specify every aspect of deployment.

The move has also been a shift from a heavyweight application deployment cloud model to a lightweight cloud native model that takes less time to start up and deploy applications. Additionally, there has been a move from being bound to a single, closed-source vendor to an open-source model with multiple vendors, resulting in less risk of lock-in and more choice. All the major cloud providers such as AWS, Azure, Google Cloud Platform, etc. have adopted this cloud native approach. In this module we will learn about the history, current status and future of cloud native computing.

1B

Introduction to Linux and Bashusing Python

Linux containers are poised to take over the world; we will start our course with an introduction of Linux and the comm and line. For many non-technical peopl e, the command line (also referred to as CLI, Terminal, bash, or shell) is a place of mystery. Howe ver, you only have to know a handful of basic comman ds to start feeling comfortable. In this module we will c over the basic commands to get you started.

1C

Docker Deep Dive

This course provides a soup-to-nuts lear ning experience for core Docker technol ogies, including the Docker Engine, Ima ges, Containers, Registries, Networking, Storage, and more. All of the behind th e scenes theory is explained, and all concepts are clearl y demonstrated on the command line. No prior knowle dge of Docker or Linux is required.

1D

JavaScript/TypeScript Programming

JavaScript is the default language of th e web and the first universal programm ing language. In this module, students will dig through all the genuinely elega nt parts of JavaScript, including syntax, objects, functions, inheritance, arrays, methods, etc. Thi s module will build a solid foundation in functional and object-oriented programming using JavaScript (ES 201 6) and TypeScript using Node.js. Assuming no prior pro gramming experience on the part of the student, this module starts by focusing on the fundamentals of com posing code with JavaScript. Introduction to only serve r based JavaScript/TypeScript is covered i.e. Node.js. Stu dents will also learn about the syntax and structure of the language including operators and expressions, con trol structures, methods, and arrays using Node.

2A

Developing Microservices using Node.js and Express

This module includes how to build microservices using Node.js. It starts by introducing Node's powerful traits and shows students how they map to the features of microservices using JavaScript. Students will explore key development techniques, meet the rich ecosystem of companion tools and libraries, and get a glimpse into its inner workings. In recent years, REST (REpresentational State Transfer) has emerged as the standard architectural design for web services and web APIs. In this module, students will see how easy it is to create a RESTful web service using Node.js, TypeScript and the Express micro-framework. Students will also learn gRPC and GraphQL based microservices. By the end of the module, students will be able to use JavaScript/Typescript to build a Node microservice and know how to test it, hook it up to a database, and automate the development process.

2B

Version Control with Git

You won't find a top programmer, cloud developer, or AI engineer who doesn't use version control. This is true because version control helps developers produce better results and makes collaboration easy. Around the world, in teams both large and small, Git is an essential part of the toolchain. This module covers Git and Github.

2C

Cloud Automation using Ansible

Ansible is an open-source IT automation engine which can remove drudgery from work life and will also dramatically improve the scalability, consistency, and reliability of your IT environment. In this module, students will start to explore how to automate repetitive cloud administration tasks using Ansible.

3A

Kubernetes in Practice

The Kubernetes container orchestration system safely manages the structure and flow of a distributed application, organizing containers and services for maximum efficiency. Kubernetes serves as an operating system for your clusters, eliminating the need to factor the underlying network and server infrastructure into your designs. In this quarter, students will learn to use Kubernetes to deploy container-based distributed applications. The module will start with an overview of Docker and Kubernetes before building the first Kubernetes cluster. We'll gradually expand the student's initial application, adding features and deepening knowledge of Kubernetes architecture and operation. Students will explore high-value topics like monitoring, tuning, and scaling. By the end of the quarter, students will be able to appear in the Certified Kubernetes Application Developer (CKAD) exam.

4A

Amazon Web Services Application Development

In this quarter, students will develop an understanding of core AWS services, uses, and basic AWS architecture best practices. Students will develop proficiency in developing, deploying, and debugging cloud-based applications using AWS. Students will also develop the ability to use the AWS service APIs, AWS CLI, and SDKs to write applications and gain an understanding of the AWS shared responsibility model. Students will also focus on understanding of application life cycle management, ability to use a CI/CD pipeline to deploy applications on AWS, and ability to write code using AWS security best practices. Proficiency, writing code for serverless applications, and understanding of the use of containers in the development process will also be covered in detail. By the end of the quarter students will be able to appear in the AWS Certified Developer – Associate exam.