**Training Report**

The topics that were covered in the external training sessions are:

1. Linux/Unix
2. OS fundamentals
3. Networking
4. AWS cloud services
5. Secure App Development
6. SDLC
7. Spring Boot
8. HTML/CSS JS
9. Hadoop
10. Kafka

**LINUX**:

Trainer taught us useful Linux commands during these sessions. We were given a list of commands that we would perform on our systems and later we would revise it by sharing our screens to the lecturer.

Overall the sessions were helpful and informative.

**Networking**:

Many useful topics like DNS lookup, Ethernet ARP, Ip Address and Subnetting,

Default gateway, firewalls, LAN and WAN, Routers, Switches, OSI Model, TCP/IP model Network Topology, Network Security, Different Layers, TCP vs UDP , HTTP and HTTPs.

Also practical work was given to explore networking tools like ping, netstat, ipconfig, Wireshark.

**AWS Cloud Services**: We were provided with credentials for AWS to explore the platform and perform tasks on it. Overall the lectures were informative and taught concepts like EC2, S3, VPC, EBS, learned how Identity and Access Management can be used to handle security in AWS cloud environments.

Learned how to create our own database.

Learned various storage service provided by AWS and gets hands-on experience with both amazon S3 and Amazon Elastic Block Store(EBS).

Understand the compute services available in AWS and how to launch an EC2 instance.

**Secure App Development**:

In these sessions we were taught about different security vulnerabilities that a programmer must know so that attackers can’t easily do damage to our data.

We were taught about:

* Cross site scripting
* Man in the middle Attacks
* Phishing Attacks
* Implementing server side checks to reduce SQL injection.
* Usage of APIs

**SDLC**:

In the software development life cycle we were steps of the lifecycle of creating a product we were also given an assignment on this particular topic. SDLC of a e-commerce website:

**Phase 1**: Planning: - Target Audience: what type of customers do you want to target - Order, Account creation, Delivery Status - Website, App which to create first

**Phase 2** : Design - Which tech stack to use - UML diagram, database schema designing - Identifying the flow in which user will interact with the product - Method of payment - Way to register a user.

**Phase 3**: Implementation - Coding the project for the planned iteration - Making the code as modular as possible - identifying routines which can be reduced to various subroutines.

**Phase 4**: Testing - Testing the coded software in various iterations: alpha, beta, etc. - Doing unit, integration, performance testing.

**Phase 5**: Maintenance - Making the app scalable for millions of users - If required going back a step in the life cycle.

**SpringBoot:**

We were taught a very popular and widely used open source based backend service Spring Boot. We were taught the proper architecture while creating a project (MVC). A project was also based on this topic. We were asked to create the backend taking production into account with authentication using JWT and a database using SQL.

**Unit Testing:**

Unit testing is a software development process in which the smallest testable parts of an application, called units, are individually and independently scrutinized for proper operation. This testing methodology is done during the development process by the software developers and sometimes QA staff. It can be done manually but usually automated. The internal work I did in my team here at ZScaler was also on unit testing so I was able to understand the use and necessity of unit testing.

**Selenium Testing**:

Selenium is an open source umbrella project for a range of tools and libraries aimed at supporting browser automation. It provides a playback tool for authoring functional tests across most modern web browsers, without the need to learn a test scripting language. In the backend we developed in Spring Boot lectures, we did Selenium testing on that.

**Docker**:

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime. Using Docker, you can quickly deploy and scale applications into any environment and know your code will run.

**Kafka & Hadoop:**

The topics that were covered in the session are Introduction to Kafka, Architecture, Kafka Consumers and Producers, Kafka Broker, Uses of Kafka, Big Data, Map Reduce, HDFS File, system, Data nodes, Name nodes. We also Executed a Java Program to implement Map Reduce functionality of Hadoop in Java.