**Akka Framework Training**

|  |  |
| --- | --- |
| Summary | |
| Scope | Akka |
| Language | Java/Scala |
| Pre-requistes for Developers | Strong knowledge on Core Java  Strong knowledge on JVM  Working with Jars, classpaths etc |
| Objective | Getting developers hands-on with Akka, Concurrent programming, Actor models, Akka cluster, Routing, Mailbox, Fault Tolerance, FSM and Persistence, Deployment |
| Scala Version | 2.12 |
| Days | 3 days [More hands-on] using Scala or Java |
| Add-on Days | 1 or 2 additional days if Scala training required. |
| Hands-on | 75% |
| Presentations,  Fundamentals, Demos | 25% |

**Course ToC**

|  |  |
| --- | --- |
| **Day 1- Basic and Intermediate Part** | |
| Introduction  Ramp up developers on problem statement  Concept | Messaging Introduction  Actor model Introduction  Concurrency, Parallel processing, threading  Pains with Locks  Sync vs Async model  Blocking vs Non-Blocking models  Blocking vs Non-Blocking IOs  Race Conditions  Deadlocks  Compare messaging with AMQP/Kafka  Compare actor with Apache Spark/Map Reduce  Messaging vs Streaming  Mailbox |
| Setup  Hands-on | Setup JDK  Setup Build environment Maven for Java/SBT for Scala  IDE and Tools setup  Development Environment |
| Akka Introduction  Concept | Introduction to Akka Framework  Understanding Akka concurrency models and architecture  Akka features: Concurrency, Stability, Fault Tolerant, Cluster |
| Akka Architecture  Concept | Akka Architectures  Elements of Akka framework and Runtime |
| Actor  Hands-on | Actor Introduction  Hello world with Akka framework with Actor  Actor Architecture  Actor with Props  Talk to Actor  ActorRef  Actor Messaging  Send message to Actor  Get Reply from Actor |
| Actor Hierarchical Structure  Hands-one | Creating Child actors  Actor Hierarchical Structure |
| Actor Life Cycle  Hands-on | Actor Life Cycle  With Hierarchical Structure  Stoping, Killing, Poisioning Actor |
| Actor System | Stopping, terminating actor system,  Root actor names  System context  Actor context |
| Messaging with Akka  Hands-on | Actor Path, Actor Selection  Selecting Actor using path  Sending message to all siblings  Sending messages to parent  Dispatching messages |
| Supervisor and Fault tolerant  Hands-on | Supervisor Actors Introduction  Supervisor Strategy  OneForOne  OneForAll strategy  Using BackoffSupervisor  Customizing Supervisor Strategy |
| Actor State  Hands-on | Understanding Actor State  Recovering Actor state on failure strategy  Stateless Actors |
| Actor Behaviour  Hands-on | Changing Actor Behaviour  Becoming and unbecoming behaviour  The need for behaviour change use cases |

|  |  |
| --- | --- |
| **Day 2 –Akka (Intermediate)** | |
| Recap Akka Basics discussed on Day 1  Concept | Akka and Actor model  Supervisor Strategy  Actor Behavior Become/unbecome  Messasing |
| Config  Hands-on | Akka configuration basics  Config files  Reading config files |
| Mailbox  Hands-on | Akka Mailbox  Bounded Mailbox  Priority Mailbox  Queue model  Messaging Configuration |
| Dispatchers  Hands-on | Akka Dispatchers  Default, Pinned and CallingThread Dispatchers |
| Router  Hands-on | Akka Router mechanism  Creating Child Routes  Routes termination and recreation  Sending Messages via Router |
| Akka FSM  Hands-on | Finite State Machine Introduction  Using Akka for FSM |
| Akka Persistence  Hands-on | Persistent Actors  Persistent FSM  Persistent Query |

|  |  |
| --- | --- |
| **Day 3- Akka Advanced Part** | |
| Recap | Day 1 Recap  Day 2 Recap |
| Cluster  Hands-on | Clustering Introduction  Working with Remote Actor  Akka Cluster  Building Cluster  Cluster Singleton  Docker or VM required, if not different port numbers used on same host machine |
| Cluster Sharding  Hands-on | Introduction and configuration |
| Akka Stream  Hands-on | Akka Stream  Reactive Tweets  Stream IO examples |
| Akka Connectivity  Hands-on | Akka HTTP APIs  Akka HTTP Client  Akka HTTP Server/Client example  MySQL Example |
| Akka Design Patterns  Concept and Demo | Loading Balancing  Throttling Messages  Shutdown Patterns  Ordered Termination  Scheduled messages |
| Fault Tolerance with Actors  Concept | Achieving Fault tolerance with Akka Best practices |
| Performance  Concept and Demo | Tuning an Akka system  Monitoring and reporting |
| Akka for Production  Hands-on and Concept | Bundle Akka for Production  Deployment scenarios |