

# Continuous Delivery for Microservices with SBT and Docker

Patterns behind a mnubo internal revolution



# Continuous Delivery





# Docker



# Company SBT plugin

```
object MnuboApplicationPlugin extends AutoPlugin {  
  
  private val artifactsRepo = "Mnubo releases" at s"http://some-url"  
  
  override lazy val requires = DockerPlugin && AssemblyPlugin && ReleasePlugin  
  
  override lazy val projectSettings = Seq(  
    organization           := "com.mnubo",  
    scalacOptions          += Seq("-deprecation", "-feature"),  
    resolvers              += artifactsRepo,  
    publishTo              := Some(artifactsRepo)  
    // ...  
  )  
}
```



# Latest plugin

```
val pluginArtifactUrl = new URL(
  "<artifactory URL>/com/mnubo/mnubo-plugin_2.10_0.13/maven-metadata.xml"
)

val latestPluginVersion =
  Using.urlInputStream(pluginArtifactUrl) { stream =>
    """"<latest>([\d\.]+)</latest>""".r
      .findFirstMatchIn(IO.readStream(stream))
      .get
      .group(1)
  }

resolvers += "Mnubo repository" at "<artifactory URL>"

addSbtPlugin("com.mnubo" % "mnubo-plugin" % latestPluginVersion)
```

# Automating the build

```
releaseVersion          := identity,  
releaseNextVersion      := { (ver: String) =>  
  sbtrelease  
    .Version(ver)  
    .map(_ . bumpBugfix.string)  
    .getOrElse(sys.error("No versions are set!"))  
},  
releaseProcess          := Seq[ReleaseStep](  
  setupRemoteTracking,  
  inquireVersions,  
  runTest,  
  setReleaseVersion,  
  publishArtifacts,  
  tagWithOverwriteRelease,  
  setNextVersion,  
  commitNextVersion,  
  pushChanges  
)
```



# Building Docker images

# sbt-native-packager



# sbt-docker





# Latest version

```
libraryDependencies += Seq(  
  "com.mnubo" %% "some-lib" % "[1.0.0,)" changing()  
  //...  
)
```



# Managing data stores



<https://github.com/mnubo/dbevoln>



# Contact and questions



julien.letrouit@gmail.com



@jletroui