Writing lamba function

Handler – write a main method starting point of lamba no business logic

Controller – Event processing core business logic

Service – external integration services

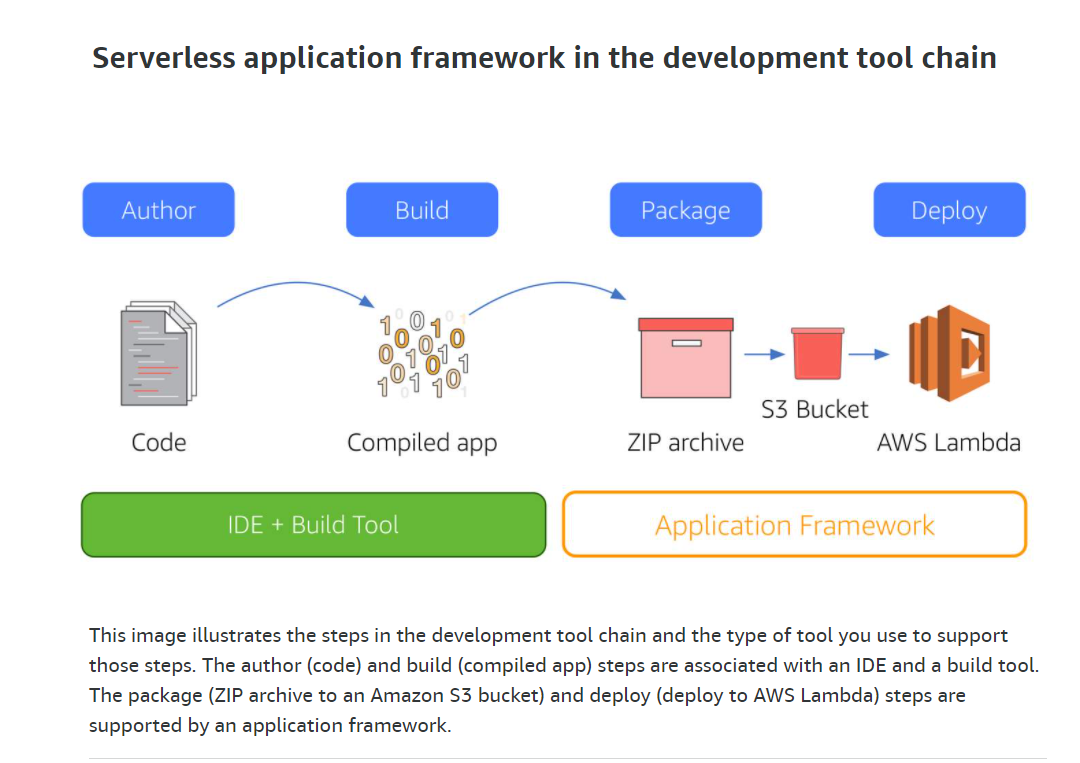
Managing Serverless application:

Traditional application had scripts which use maven and gradle to re-create the binaries in the server .

With serverless application the ie maven and gradle complied binaries will be packaged ie zipped and then deployed on S3 using AWS Lamba function.

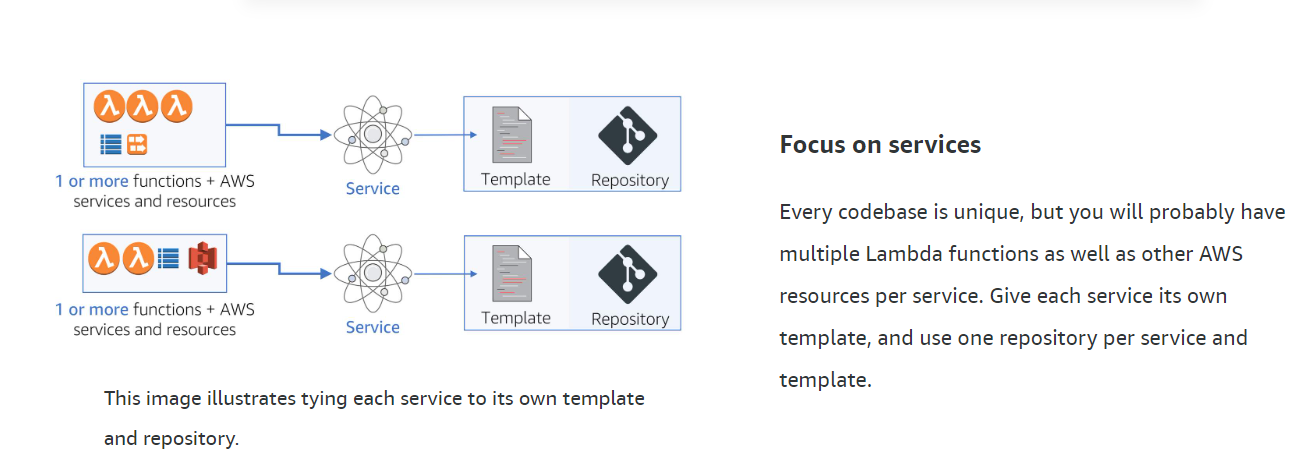
The packaging and deploying will be in SAM framework

You don't need to switch your familiar developer toolchain to go serverless, but you should use a serverless application framework to simplify the deployment process. The AWS serverless framework, SAM  uses SAM templates which are streamlined versions of an Amazon CloudFormation template.



How to organize your repo?

Each lambda service will have a template and a separate repo.



Although you may have multiple functions per service, a good organizational rule of thumb is to have one template per service and one code repository per template.

AWS Serverless Java Container For Spring boot java application

AWS Serverless Express Framework  and Zappa framework for Python etc.

For debugging AWS lamba function

Use IDE debugger tool on Docker container to intereatively debug the perform same

i.e either request something from browser or run automated unit test case against the application running locally to gain access against the function you want to debug.

The IDE debugger tool will use standard protocols and ports to connect. You don’t have to open like how we do for Solr . AWS doesn’t support open port.

**USE IDE DEBUGGER TOOL ON DOCKER CONTAINER PROVIDED BY SAM CLI (Serverless application model . Install SAM CLI)**

You cannot connect remotely to Lambda via an open port, but you can install SAM CLI and use your IDE debugger on the docker container that it provides.

<https://aws.amazon.com/serverless/>