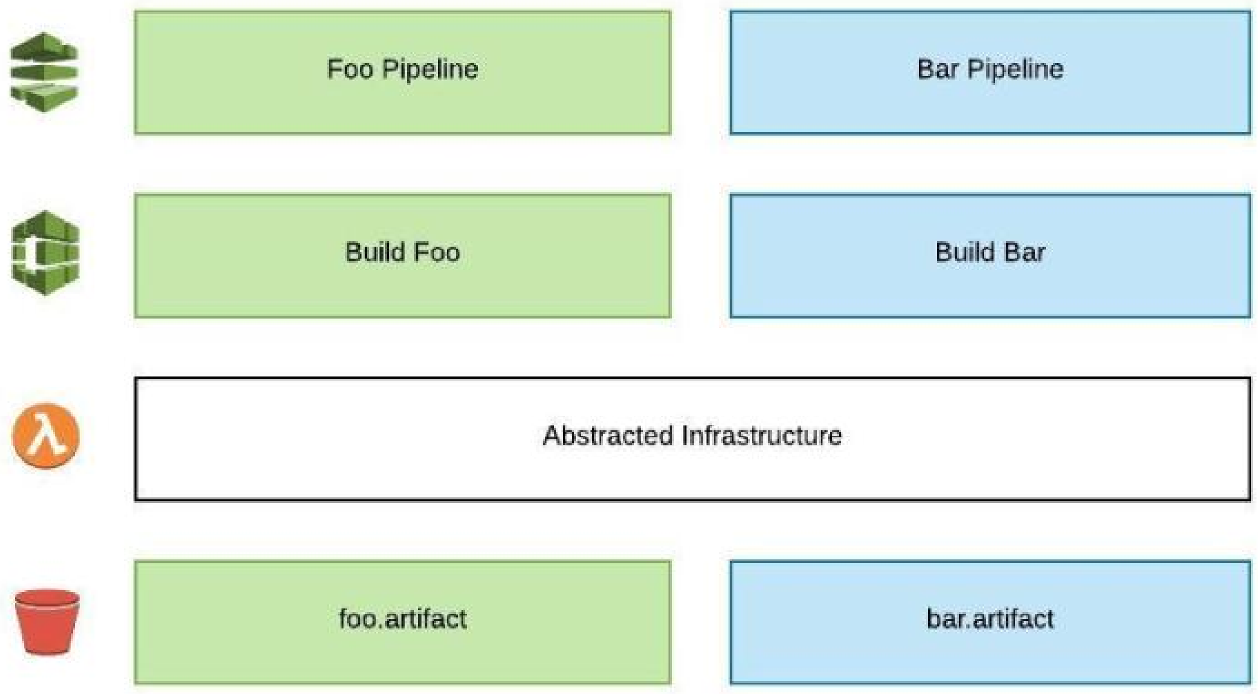
This is a continuation of the **Four Factors of an Effective Software Development Pipeline**, where we will focus on factor three: **Pipeline Maintenance Should be Free!** Check out Factor 2: Skip the Gatekeeper if you missed it.

Factor #3: Pipeline Maintenance Should be Free!

At this stage of evolution of software engineer, particularly as *agile development* is the effective standard, there cannot be a more absurd norm than the interruption of builds and deployments because the associated servers/containers need to be updated and/or patched.

Abstracted Build Infrastructure

With Lambda, or any serverless implementation, you're abstracting the execution of the builds to whatever managed service you prefer. That means there is no longer any need to patch *jenkins* build servers; there's never an interruption to your builds/pipelines unless the pipeline itself is being altered; or maybe not!

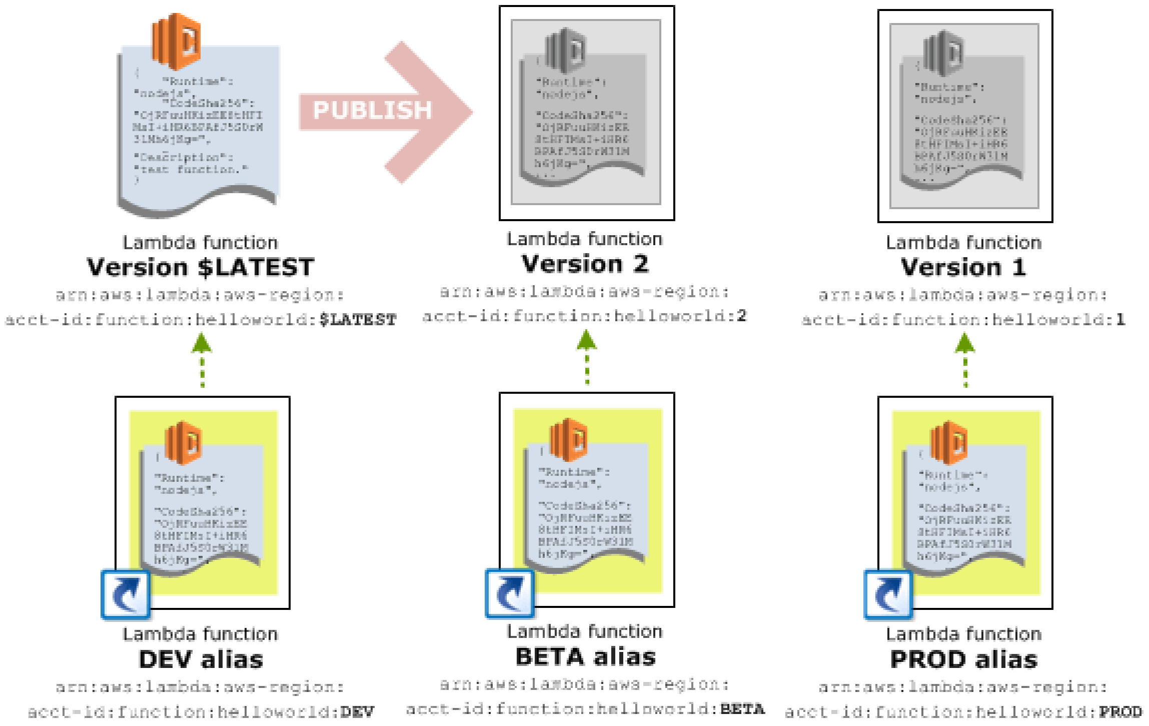


Let your cloud provider manage the infra so you can focus on delivering to your customers!

Take for example, the complexity map of plugins for Jenkins, and imagine the headache of managing an update to the Jenkins binary. The madness of maintaining Jenkins builds by the binary necessary to execute a plugin *needs to end!*

Version Your Build

AWS Lambda supports versioning and aliases on functions, which essentially gives you the power to blue/green your build scripts/functions. With this, your build functions on Lambda can be updated **without interrupting** your build. Pipelines can then reference an alias of a function by name, and you can freely move in/out versions of functions with changes you want to make without catastrophic failure. This is particularly useful for step functions.


PUBLISH
Lambda function
Version $LATEST
arn:aws:lambda:aws-region:
acct-id:function:helloworld:$LATEST
Lambda function
Version 2
arn:aws:lambda:aws-region:
acct-id:function:helloworld:2
Lambda function
Version 1
arn:aws:lambda:aws-region:
acct-id:function:helloworld:1
Lambda function
DEV alias
arn:aws:lambda:aws-region:
acct-id:function:helloworld:DEV
Lambda function
BETA alias
arn:aws:lambda:aws-region:
acct-id:function:helloworld:BETA
Lambda function
PROD alias
arn:aws:lambda:aws-region:
acct-id:function:helloworld:PROD
Borrowed from the official AWS Documentation

AWS Lambda Versioning Strategies by Kevin Ng, and Versioning in AWS Lambda by Abraham Bae are both fantastic resources that go into much better detail on this topic.

Wrapping Up

When you let someone else manage the complexity of server hardware you and your team can focus on delivery for your customers and not on troubleshooting plugins, binaries, and dependency maps of both across multiple physical or virtual servers while juggling an active set of builds by your team. Whew, that’s a mouthful, and you can avoid it all.

Recapping

Previously we discussed the second factor, which is to Skip the Gatekeeper.