

◀ Kotlin 1.4 Online Event

Kotless Kotlin Serverless Framework Vladislav Tankov



@vdtankov

October 15, 2020

Introduction to serverless

A few words on the hottest topic

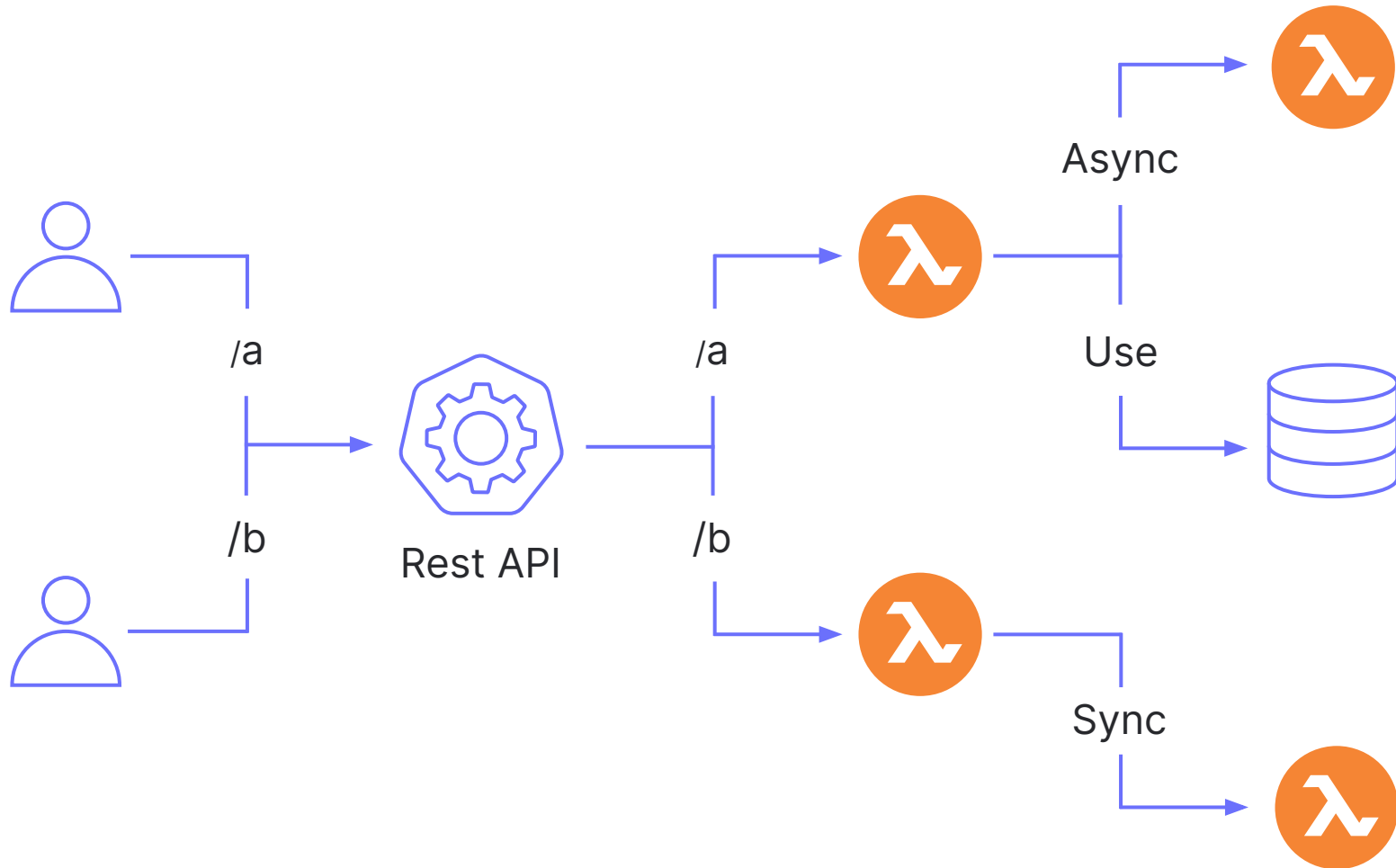
Serverless is

a cloud-computing execution model in which the cloud provider runs the server and dynamically manages the allocation of machine resources

Wikipedia

Basically, it means

1. Take small elements – stateless functions
2. Combine them with events to make an application
3. Run the application in the cloud



Why?

Pros of serverless

- Pay per request
- Scale up to thousands of CPUs
- Fault-tolerant

No cons?

Tons of configuration!

```
resource "aws_lambda_function" "_long" {
  function_name = "Handler__long"
  s3_bucket = "${aws_s3_bucket.ktlst_lambda_s3.bucket}"
  s3_key = "${aws_s3_bucket_object.ktlst_bucket_object.key}"
  source_code_hash = "${base64sha256(file("../build/libs/kotless
  handler = "kotless.Lambda::handleRequest"
  runtime = "java8"
  timeout = 30
  role = "${aws_iam_role.ktlst_lambda_role.arn}"
  memory_size = 256
  environment = {
    variables = "${var._long_envvars}"
  }
}
```

```
resource "aws_lambda_permission" "_long" {
  statement_id = "AllowAPIGatewayInvoke"
  action = "lambda:InvokeFunction"
  function_name = "${aws_lambda_function._long.arn}"
  principal = "apigateway.amazonaws.com"
  source_arn =
"${aws_api_gateway_deployment.ktlst_example_deployment.executi
}
```

```
resource "aws_api_gateway_resource" "_long" {
  parent_id = "${aws_api_gateway_rest_api.ktlst_example_rest_ap
  rest_api_id = "${aws_api_gateway_rest_api.ktlst_example_rest_
  path_part = "long"
}
```

```
resource "aws_api_gateway_method" "_long" {
  rest_api_id = "${aws_api_gateway_rest_api.ktlst_example_rest_
  resource_id = "${aws_api_gateway_resource._long.id}"
  http_method = "ANY"
  authorization = "NONE"
}
```

```
resource "aws_api_gateway_integration" "_long" {
  rest_api_id = "${aws_api_gateway_rest_api.ktlst_example_rest_
  resource_id = "${aws_api_gateway_method._long.resource_id}"
  http_method = "${aws_api_gateway_method._long.http_method}"
}
```

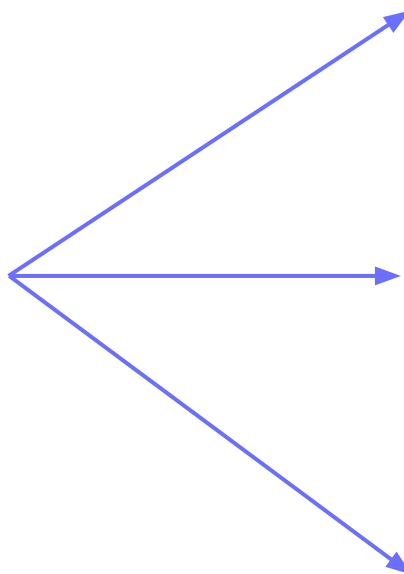
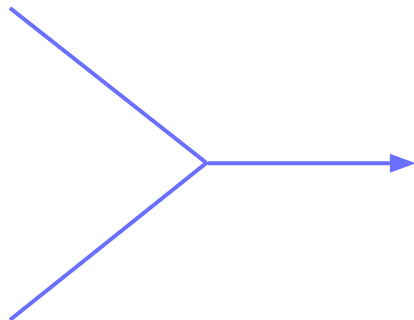
Could it be simpler?

Kotlin Serverless Framework

Let's change the game
(at least for the Web)

```
@Get("/hello-world")  
fun helloWorld(): String {  
    return "Hello World"  
}
```

Deduce configuration from code



Infrastructure in Code

- Write the code with the framework of your choice
- Choose a Cloud platform to use
- Let Kotless do the deployment for you

Code only

```
@Get("/hello-world")  
fun helloWorld(): String {  
    return "Hello World"  
}
```


What is in the box?

Current state of the project

A 3D geometric shape, possibly a pyramid or a box, rendered in shades of blue and purple, set against a dark grey background. The shape is composed of several flat, triangular and quadrilateral faces that meet at sharp edges and vertices. The lighting is dramatic, with bright highlights on the top and side surfaces, and deep shadows in the recessed areas, giving it a sense of depth and volume. The colors transition from a vibrant blue on the top and sides to a darker purple in the shadows.

What is in the box?

- Number of supported frameworks
- Each of them can be deployed to the cloud
- With the target runtime of your choice

Supported frameworks

- Ktor
- Spring Framework
- Kotless's own DSL

Supported clouds

- Amazon Web Services
- In development
 - Google Cloud Platform
 - Microsoft Azure

Supported runtimes

- Kotlin/JVM
- GraalVM
- In development
 - Kotlin/JS

Frameworks



Runtimes



GraalVM™



Clouds



And even local starts!

Local development

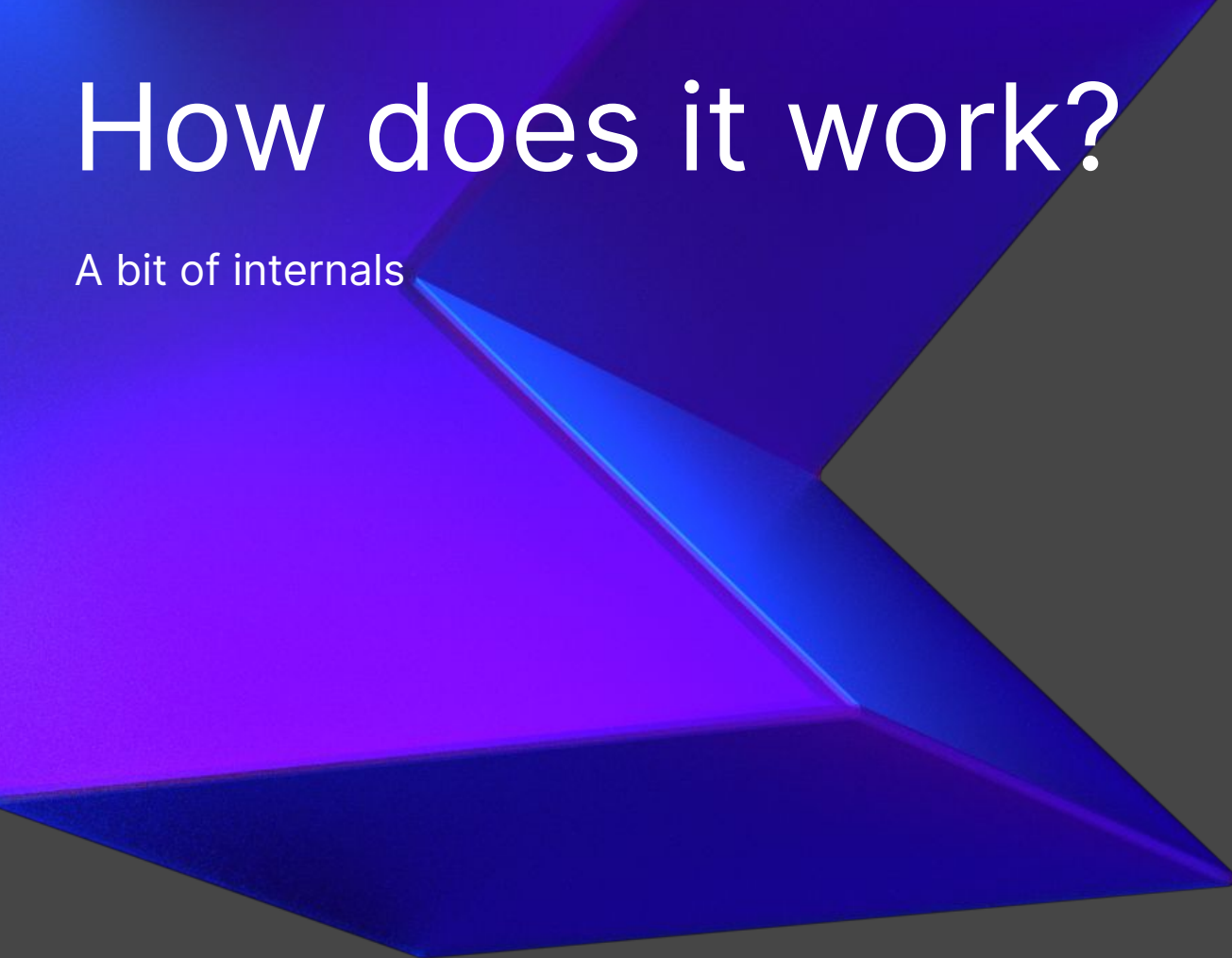
- Local emulation of Cloud services
- Ability to debug serverless applications
- No cloud account required to try it

Summing it up

- Choose the runtime that best meets your needs
- Write an application with your favorite framework
- Test and debug it locally
- Deploy an application to your pr

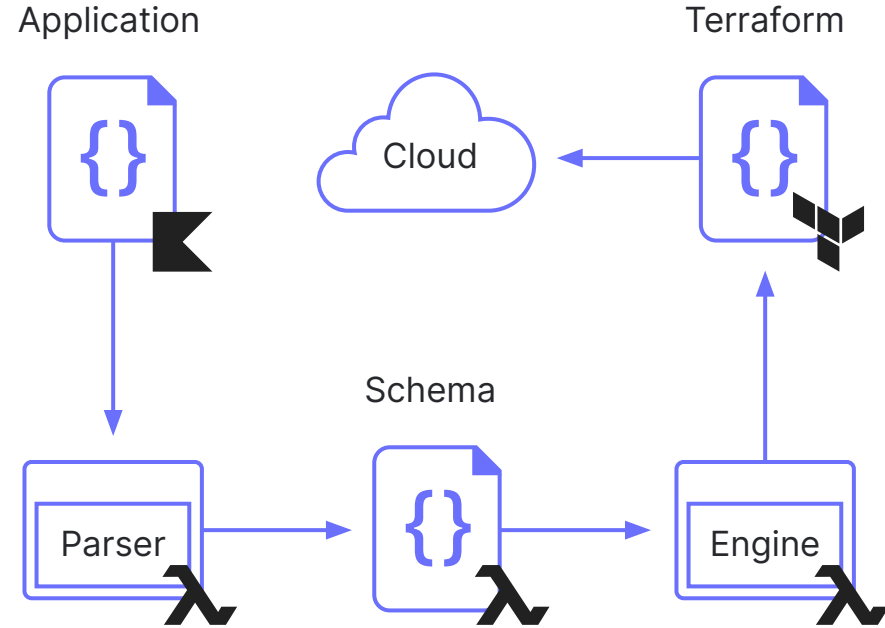
How does it work?

A bit of internals



Under the hood

- Parse an application
- Generate a cloud-agnostic Schema
- Transpile the Schema to a Terraform
- Perform the deployment



Cloud integration

- Permissions are granted via Cloud-specific annotations
- Events also work via annotations

```
@DynamoDBTable("table", ReadWrite)
object URLStorage {
    fun getByCode(code: String): String {
        ...
    }
}
```

Demo

Better than a thousand words

What is next?

Future plans



In development

- Cloud Platforms
 - Google Cloud Platform
 - Microsoft Azure
- Runtimes
 - GraalVM
 - Kotlin/JS

Future plans

- IntelliJ IDEA plugin
 - Remote logs tailing
 - Debugging
- Cloud-specific extensions
 - Authentication
 - Events
- And lots more!

Thanks!
Have a nice Kotlin



@vdtankov