Azure Functions

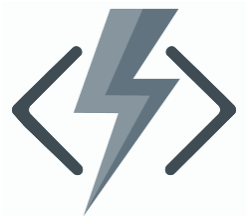
from a single Function to Durable workflow

Author: Szymon Bernad

19.12.2019

Agenda

- Brief description of Azure Functions concept
- Simple example
- Configuration and error handling
- Advantages and limitations
- Durable Functions: (a bit more complex) example
- Discussion: your experience with Azure Functions

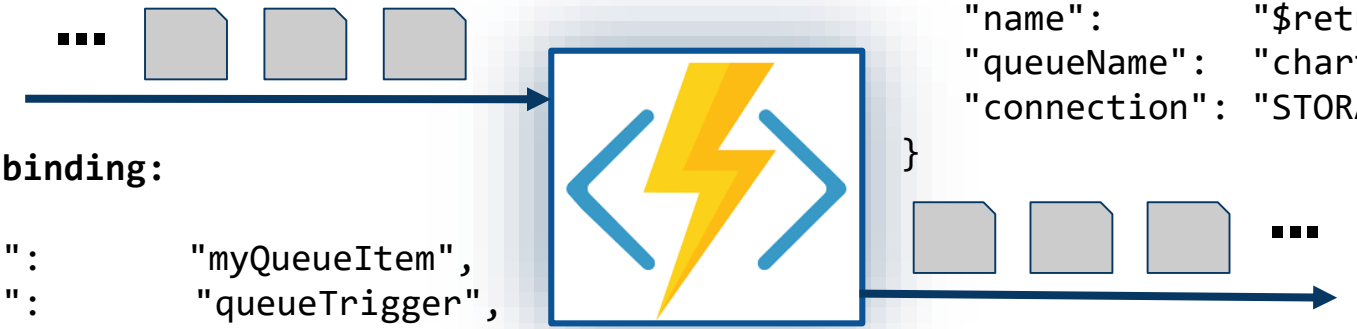


Azure... Functions?

- Lightweight application containers - part of MS Azure “serverless” platform
- Relatively short lifetime and stateless by design

... maybe let’s see an actual example

1st Example



input binding:

```
{
 "name":       "myQueueItem",
 "type":       "queueTrigger",
 "direction":  "in",
 "queueName":  "chart-requests",
 "connection": "STORAGE"
}
```

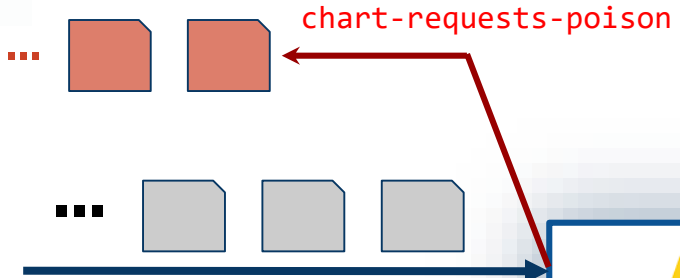
output binding:

```
{
    "type":       "queue",
    "direction":  "out",
    "name":       "$return",
    "queueName":  "chart-urls",
    "connection": "STORAGE"
}
```

Global config for function app - host.json

```
{
  "version": "2.0",
  "extensions": {
    "queues": {
          "maxPollingInterval": "00:00:02",
          "visibilityTimeout": "00:00:30",
          "maxDequeueCount": 5,
          "batchSize": 16,
          "newBatchThreshold": 8
    }
  }
}
```

1st Example



input binding:

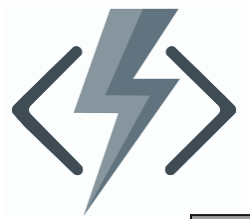
```
{
 "name":       "myQueueItem",
 "type":       "queueTrigger",
 "direction":  "in",
 "queueName":  "chart-requests",
 "connection": "STORAGE"
}
```

output binding:

```
{
    "type":       "queue",
    "direction":  "out",
    "name":       "$return",
    "queueName":  "chart-urls",
    "connection": "STORAGE"
}
```

Advantages

- Optimized for resource consumption and scalability by Azure team
- Allows for consumption-based pricing model
- Easy to use for some of common cloud development scenarios
- Built-in triggers and bindings
- Developers can focus on the actual business logic



Limitations?

Resource	Consumption plan	Premium plan	App Service plan
Scale out	Event driven	Event driven	Manual/autoscale
Max instances	**200**	**100**	**10-20**
Default time out duration (min)	**5**	**30**	**302**
Max time out duration (min)	**10**	**60**	**unbounded**

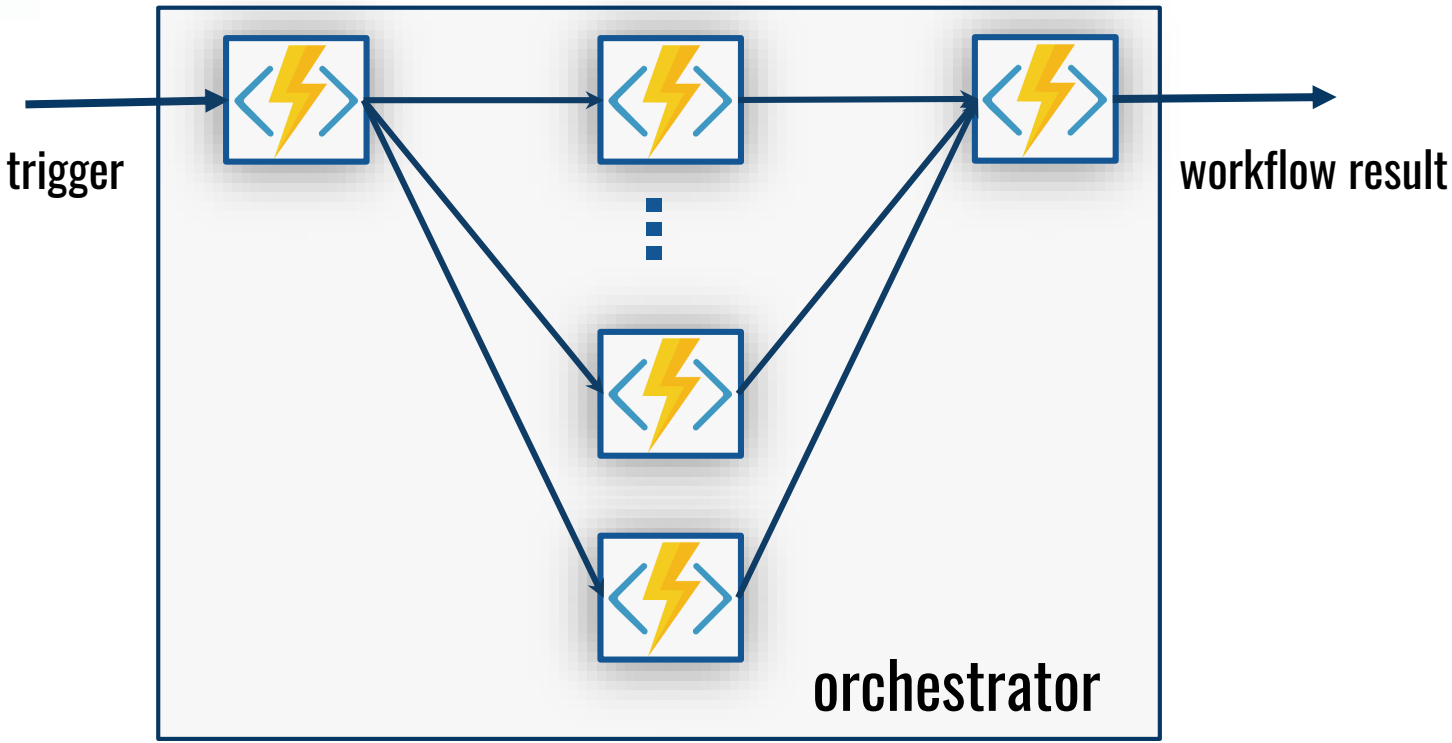
More limitations?

- Statelessness:
  *any state of function that is required to be **persistent** needs to be **externalized** outside of the instance.*

- When combining Azure Functions with high complexity -> bad design violating SRP

- Chain of functions with output-input bindings: (potentially) hard time with deployment and troubleshooting

2nd Example: durable workflow



Durable Functions

- Extension of Azure Functions - introduces orchestrator function
- Orchestrator function defines **function workflow using procedural code**.
- Orchestrator functions are durable and reliable (automatic checkpoint on each "await"/”yield”)
- Transparent “event sourcing” mechanism using Storage Account
- Workflow can be long-running (i.e.: hours, days, or even never-ending loop)

Discussion

Your experience with Azure Functions?

Useful links

- GitHub repo:
https://github.com/sim8500/AzureFunctions-TechTalk

- Docs:
https://docs.microsoft.com/en-us/azure/azure-functions/functions-bindings-storage-queue
https://docs.microsoft.com/en-us/azure/azure-functions/durable/durable-functions-orchestrations

- More general about Serverless:
https://martinfowler.com/articles/serverless.html
https://www.youtube.com/watch?v=ovRt6O7q1rA

The End

Thank you!