CNCF Serverless WG
Where do we go from here?

Doug Davis

dug@us.ibm.com
@duginabox
Agenda

• Serverless WG Overview
• CloudEvents
• Workflow
• What's new?
• Birds-of-a-feather!
• CNCF Technical Oversight Committee initiated (mid-2017)
  • Whitepaper
    • Overview of technology
    • State of ecosystem
    • Recommendations for possible CNCF next steps
  • Landscape

• Exploratory -> Actions

• First step toward interop: CloudEvents
CloudEvents - Why?

Problem

• We live in a multi-cloud and multi-service world

• Events within a cloud are well known but going across clouds need more standardization on the messages

Use cases

• How do you transit events between clouds and services?

• Be able to route events efficiently without knowing the actual payload

• Well known format for transmitting metadata about events
CloudEvents - Example

HTTP - Binary

POST /event HTTP/1.0
Host: example.com
Content-Type: application/json
c-specversion: 0.3
c-type: com.bigco.newItem
c-source: http://bigco.com/repo
c-id: 610b6dd4-c85d-417b-b58f-3771e532

{
  "action": "newItem",
  "itemID": "93"
}

HTTP - Structured

POST /event HTTP/1.0
Host: example.com
Content-Type: application/cloudevents+json

{
  "specversion": "0.3",
  "type": "com.bigco.newItem",
  "source": "http://bigco.com/repo",
  "id": "610b6dd4-c85d-417b-b58f-3771e532",
  "datacontenttype": "application/json",
  "data": {
    "action": "newItem",
    "itemID": "93"
  }
}
CloudEvents - Deliverables

- **CloudEvents Specification** – define the metadata - v0.3 released June 13
- Serialization Rules Specifications
  - **JSON** event format
  - **AMQP** event format
  - **Protobuf** event format
- Transport Bindings Specifications
  - **HTTP** – binary and structured
  - **MQTT, AMQP, NATS, Web-hooks**
  - Pointers to proprietary transport bindings
- **Primer**
- **SDKs**
  - Go-lang, Javascript, Python, Java, C-Sharp, Ruby
- **Extensions**
Serverless WG - After CloudEvents...
Pain Points

• A serverless app can be composed of a function graph with events and functions interleaved together

• The interaction between events and functions as well as how information is passed can be easily specified but no consistent way to do it
Workflow - Goals

• Often serverless functions are composed together in a chain of execution

• Define format/primitives to describe serverless application flow:
  • Includes: steps/states, event triggers (reference to events), reference to functions, how information is passed and filtered
  • Excludes: function signature, event format and metadata definition

• Facilitate portability (definition of workflow) across platforms

• Second step towards portability and interop of functions - after CloudEvents
Workflow = Execution Flow

State machine with 3 key parts:

- **Events**: Storage event, Web Request Event, etc
- **States**: event-state, operation-state, switch-state, delay-state, end-state, parallel-state
- **Actions**: functions associated with a state. Directives for parallel/sequential function execution, retry, information filtering/passing
Workflow - Status

• Working Draft **spec** - v0.1

• Workflow described as JSON document

• On-hold right now due to focus on CloudEvents
  • When CloudEvents gets to v1.0 then we'll un-pause
Proposed work streams

• APIs for accessing CloudEvents

• **Workflow (Event Orchestration / Chaining)**

• Common Function Model

• Common Serverless Benchmark framework

• Common function logging, observing and monitoring

• Function Signatures
Questions for the community... you!

Who is using Serverless in production today?
  • Main usecases / workloads?
  • What’s holding people back from using it?

What are the pain points you are experiencing w.r.t. Serverless?
  • Interop? Portability? Tooling? Debugging?

Are there workloads that aren’t well supported today but you’d like to see?

Is the split between PaaS, CaaS and FaaS/Serverless confusing? Meaningful?
Thank You

Serverless WG: [https://github.com/cncf/wg-serverless](https://github.com/cncf/wg-serverless)

CloudEvents: [https://cloudevents.io/](https://cloudevents.io/)
- Org: [https://github.com/cloudevents](https://github.com/cloudevents)
- Spec repo: [https://github.com/cloudevents/spec](https://github.com/cloudevents/spec)
- SDKs: [https://github.com/cloudevents/sdk-...](https://github.com/cloudevents/sdk-...)

Questions?