GridGain

Going Cloud-Native: Serverless Applications With Apache Ignite

Denis Magda
Apache Ignite PMC Member
GridGain Head of DevRel

Agenda

- Serverless Computing with Ignite, any profit?
- Ignite connectivity options, which one to use and when?
- Demo: Creating an Ignite serverless function
- Ignite cluster deployment, self and managed service options?

Serverless Computing With Ignite

Any Profit?

GridGain

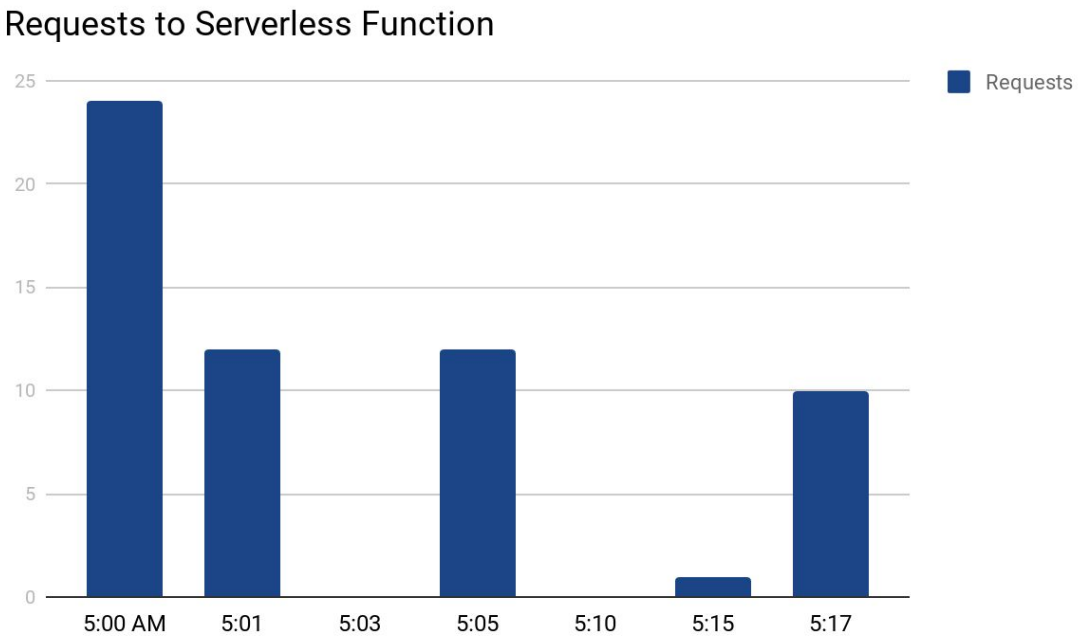
Primary Objective of Serverless Computing

cost savings

Pay for Only What You Use

What are we charged for?

- requests count
- function duration
- managed service



GridGain

Breaking Down The Duration

duration = startup_time + logic_execution_time

Breaking Down The Duration

duration = startup_time + logic_execution_time

logic_execution_time = local_logic_execution_time + remote_logic_execution_time

Ignite Decreases Logic Execution Time

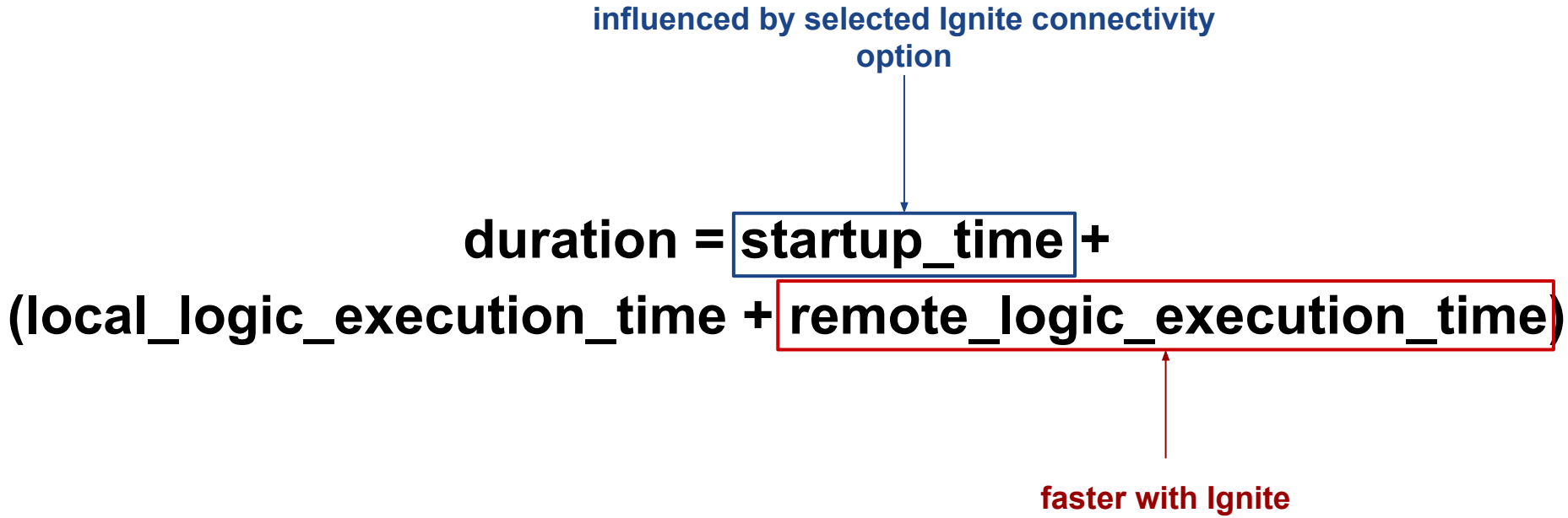
duration = startup_time +

(local_logic_execution_time + remote_logic_execution_time)

faster with Ignite

GridGain

But You Need To Select Proper Ignite Connectivity Option



GridGain

Ignite Connectivity Options

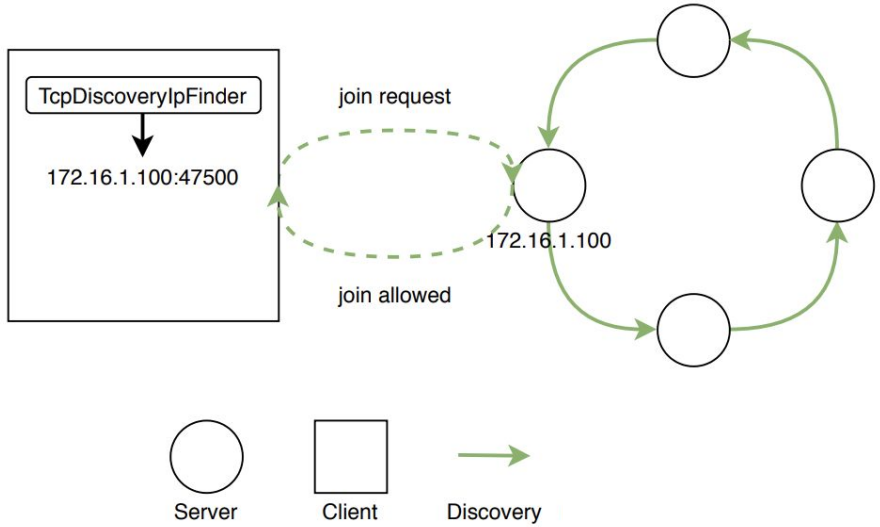
Which one to use and when?

GridGain

Thick Clients:
Not the best fit for serverless functions

- Slowest Startup Time
  - The client waits while all servers become aware of it
  - The more servers the longer the startup time

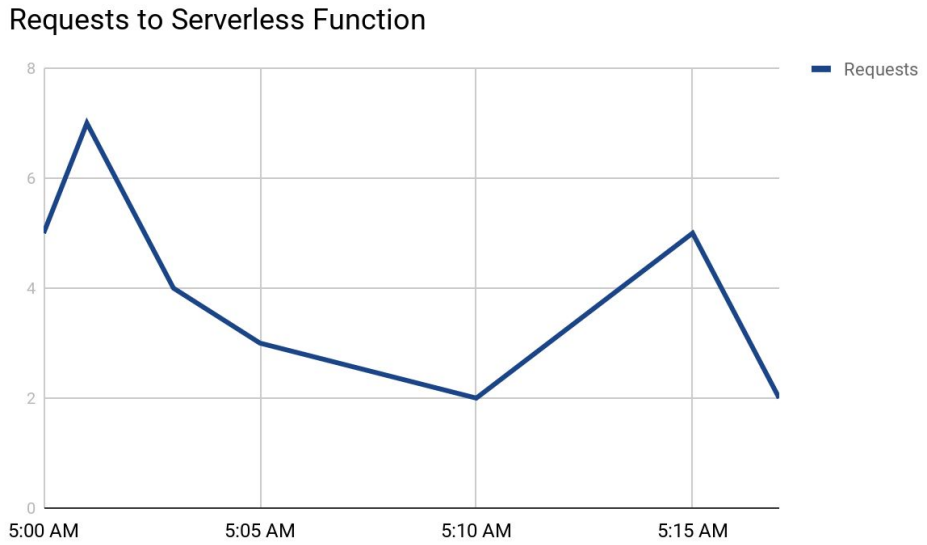
- .NET and C++ thick clients start the JVM



GridGain

Thick Clients:
A couple of reasonable usage scenarios

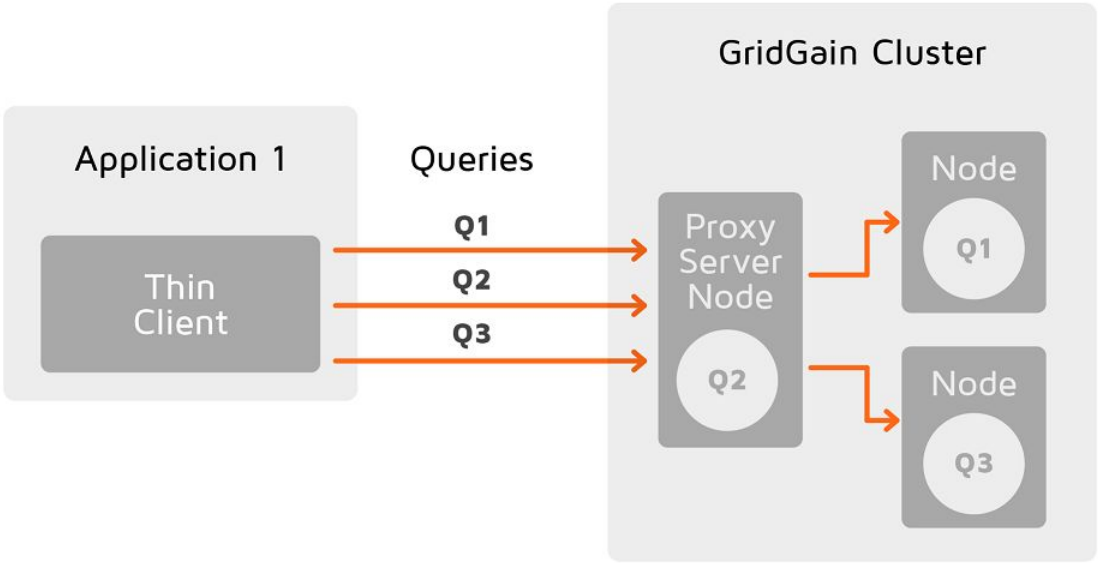
- Function traffic is consistent
  - Function is not retired/unloaded frequently
- You need an API unsupported by other connectivity options
- Ensure the client doesn't accept TCP/IP connections:
  - TcpCommunicationSpi.forceClientToServerConnections must be set to **true**



GridGain

Thin Clients
Use by default in serverless environments

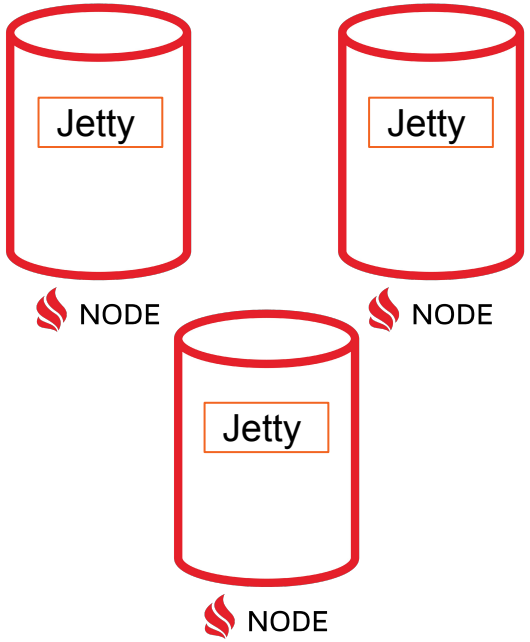
- Fast Startup Time
  - Just a TCP/IP connection with a server
- Cross-platform and lightweight
  - Java, .NET, Python, Node.js, etc.
- Feature-rich
  - Ignite 2.8: SQL, key-value, transactions
  - Ignite 2.9: compute, services and cluster APIs
  - Ignite 2.10: continuous queries



GridGain

Ignite REST Protocol:
Use to generate Graal VM native image

- Startup time is comparable to the thin client startup time
- Use with the GraalVM native image feature
  - To be supported for Ignite thin and thick clients
- Enable the **ignite-rest-http** module
  - https://www.gridgain.com/docs/latest/developers-guide/restapi



GridGain

Complete comparison of the connectivity options for serverless environments

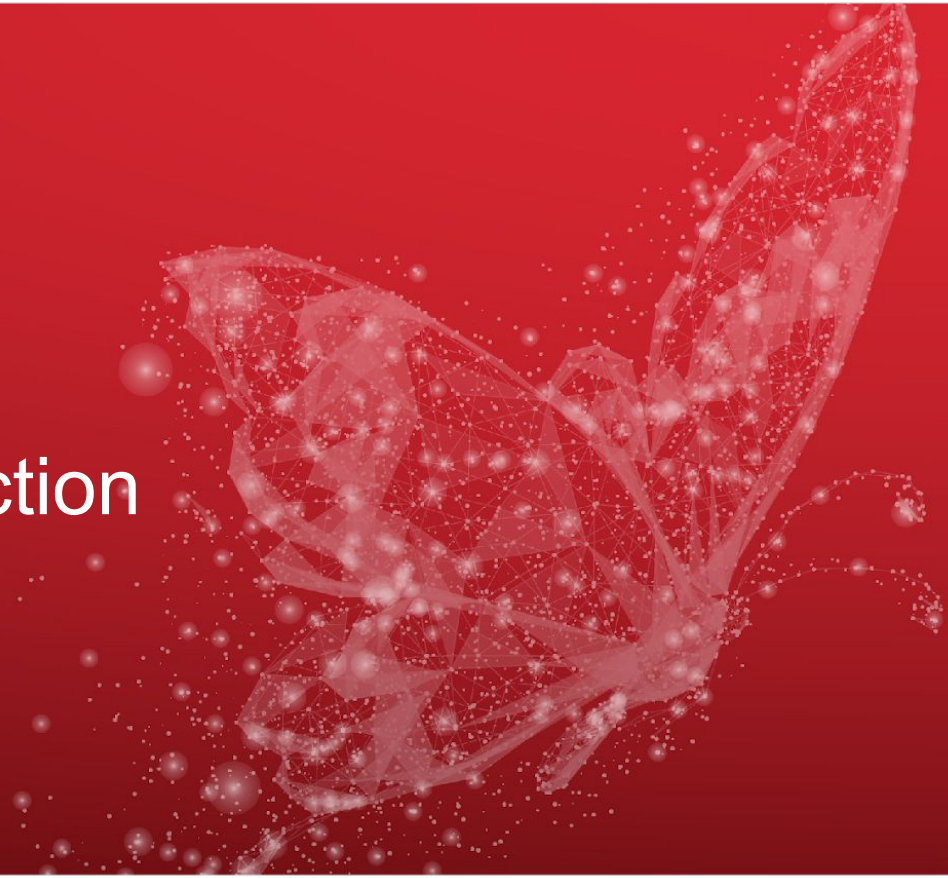
	Thin Client (+ Ignite JDBC and ODBC)	Ignite REST API	Thick Client
Startup Time	✔	✔	✘
Multi-language	✔	✔	✔ (Java, .NET, C++)
Feature Set	✔ (subset)	✔ (subset)	✔
Partition-awareness	✔	✘	✔
Graal VM Native Image	✘	✔	✘

Micronaut, Quarkus and Other Frameworks for serverless applications

- Prefer using the thin client or Ignite REST
  - To achieve fastest startup time
- Micronaut Integration
  - https://cwiki.apache.org/confluence/display/IGNITE/Micronaut+Integration
- Spring Boot integration
  - https://apacheignite-mix.readme.io/docs/spring-boot

Demo

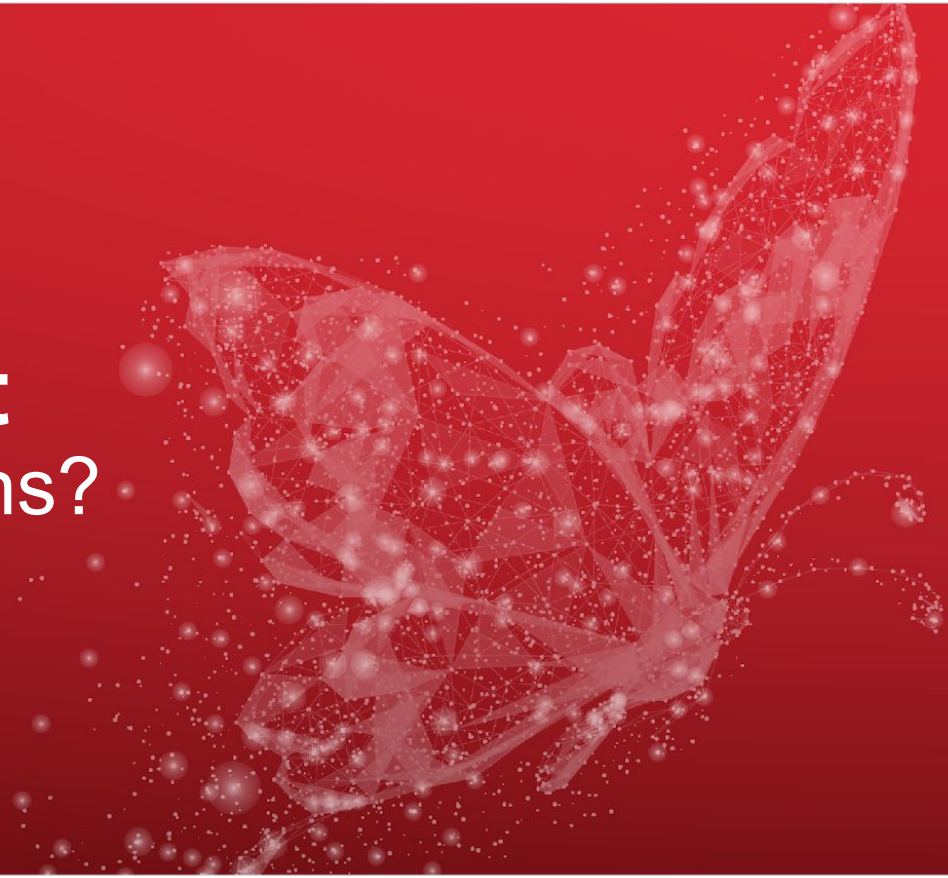
Creating an Ignite serverless function



GridGain

Ignite Cluster Deployment

Self and Managed Service Options?

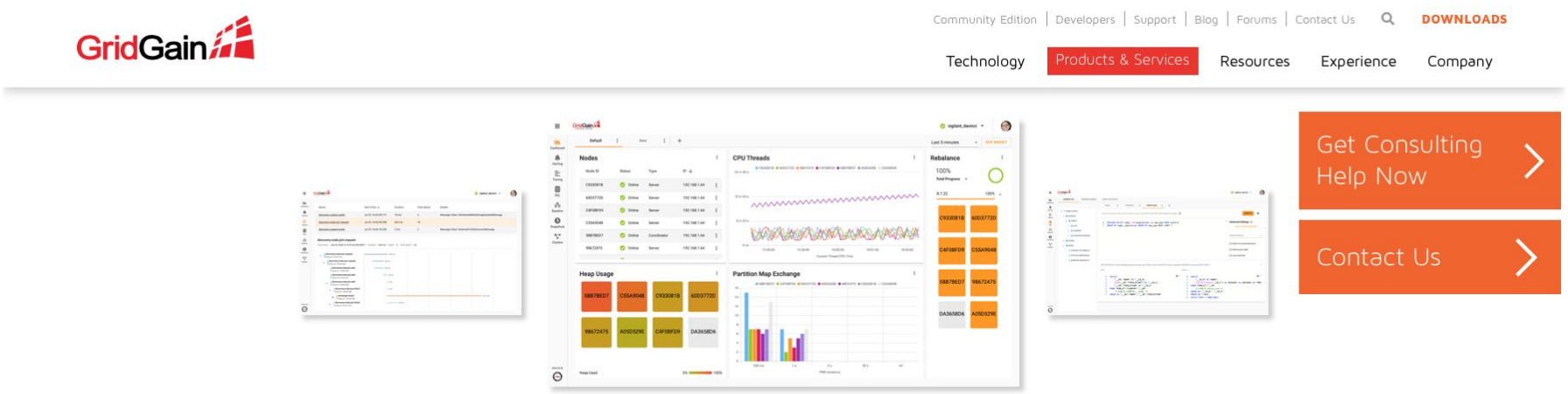


GridGain

Self-Service: Kubernetes vs VMs

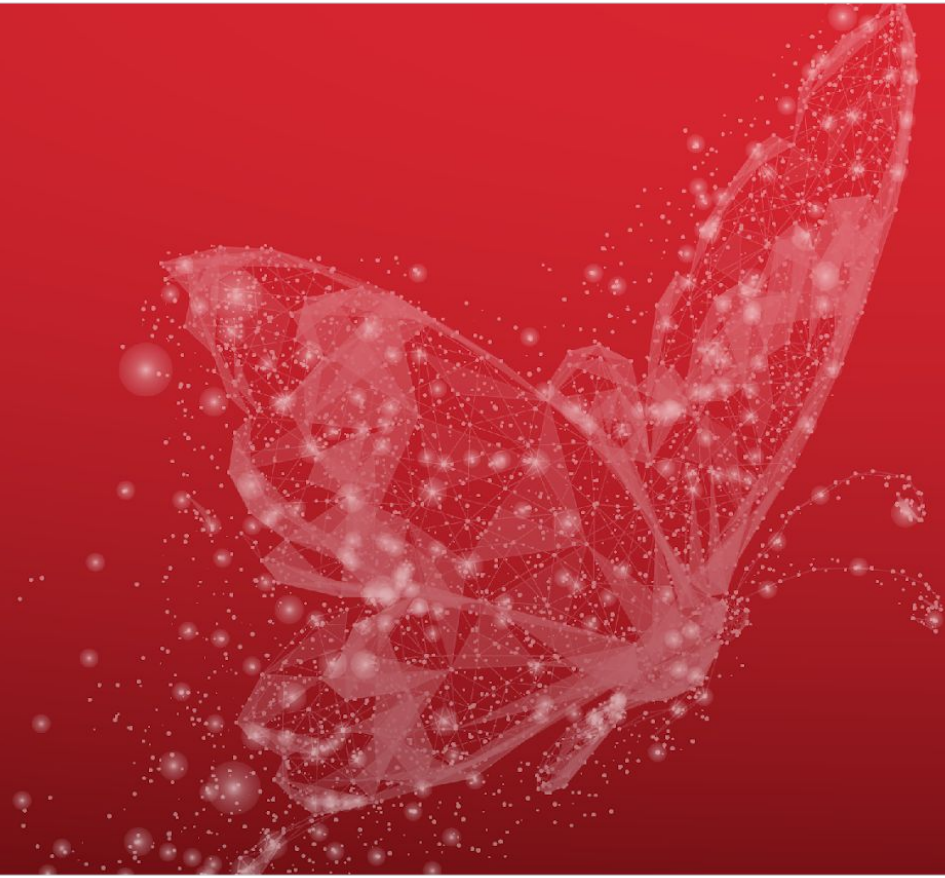
GridGain

Ignite Managed Service: GridGain Nebula



https://www.gridgain.com/products/managed-services/gridgain-nebula

Wrapping Up



GridGain

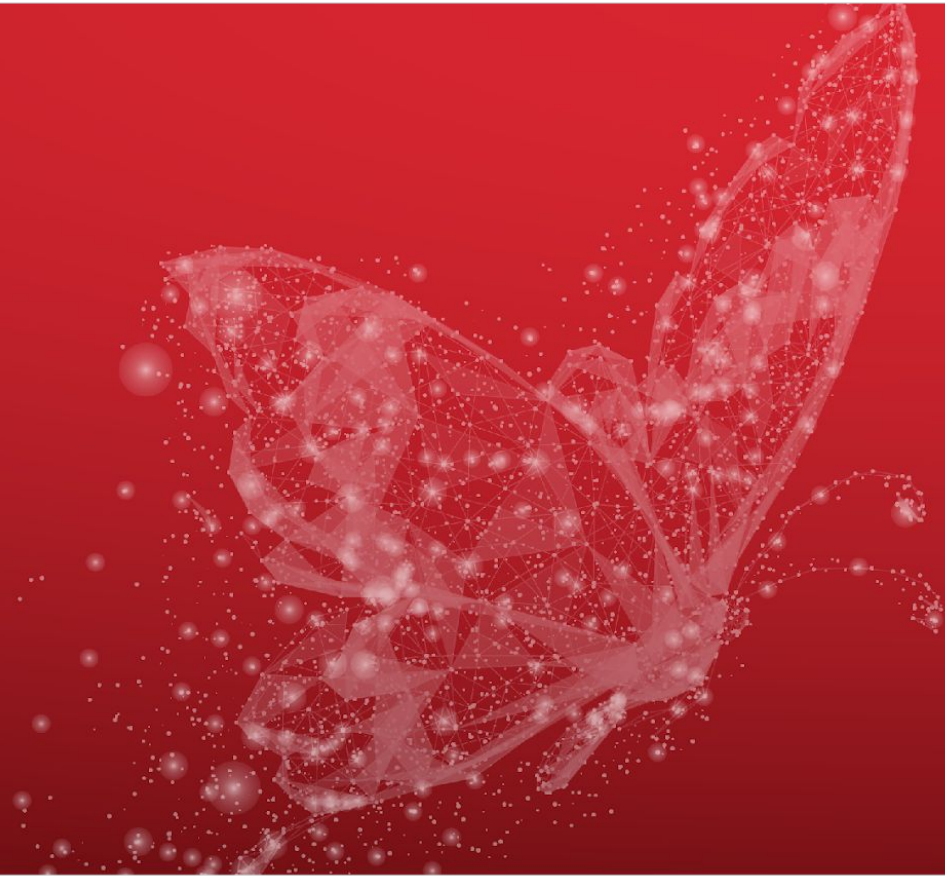
Additional Resources

- Tutorial: Deploying Ignite Serverless Functions
  - https://www.gridgain.com/docs/tutorials/serverless/azure-functions-tutorial
- Ignite with Micronaut Tutorial:
  - https://www.gridgain.com/docs/tutorials/micronaut/getting-started/ignite-micronaut-getting-started
- Serverless Architectures Deep-Dive:
  - https://martinfowler.com/articles/serverless.html#WhatIsServerless

Stay connected with Apache Ignite users & experts

meetup.com/Apache-Ignite-Virtual-Meetup/

Q&A



GridGain