Efficient Microservices Using D
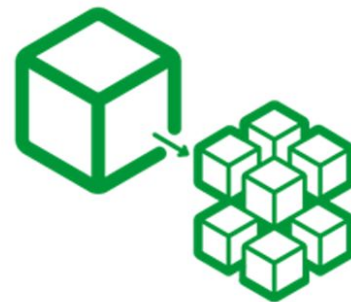
by Vijay Nayar <madric@gmail.com>
DConf 2022: August 1st

Overview

- Microservice Concepts
- Unexpected Places for System–Level Languages
- Why D for Microservices?
- Putting it into Practice
  - A Practical Problem
  - System Architecture
  - Building in the Cloud
  - Deploying in the Cloud
- Concluding Remarks

Microservice Concepts

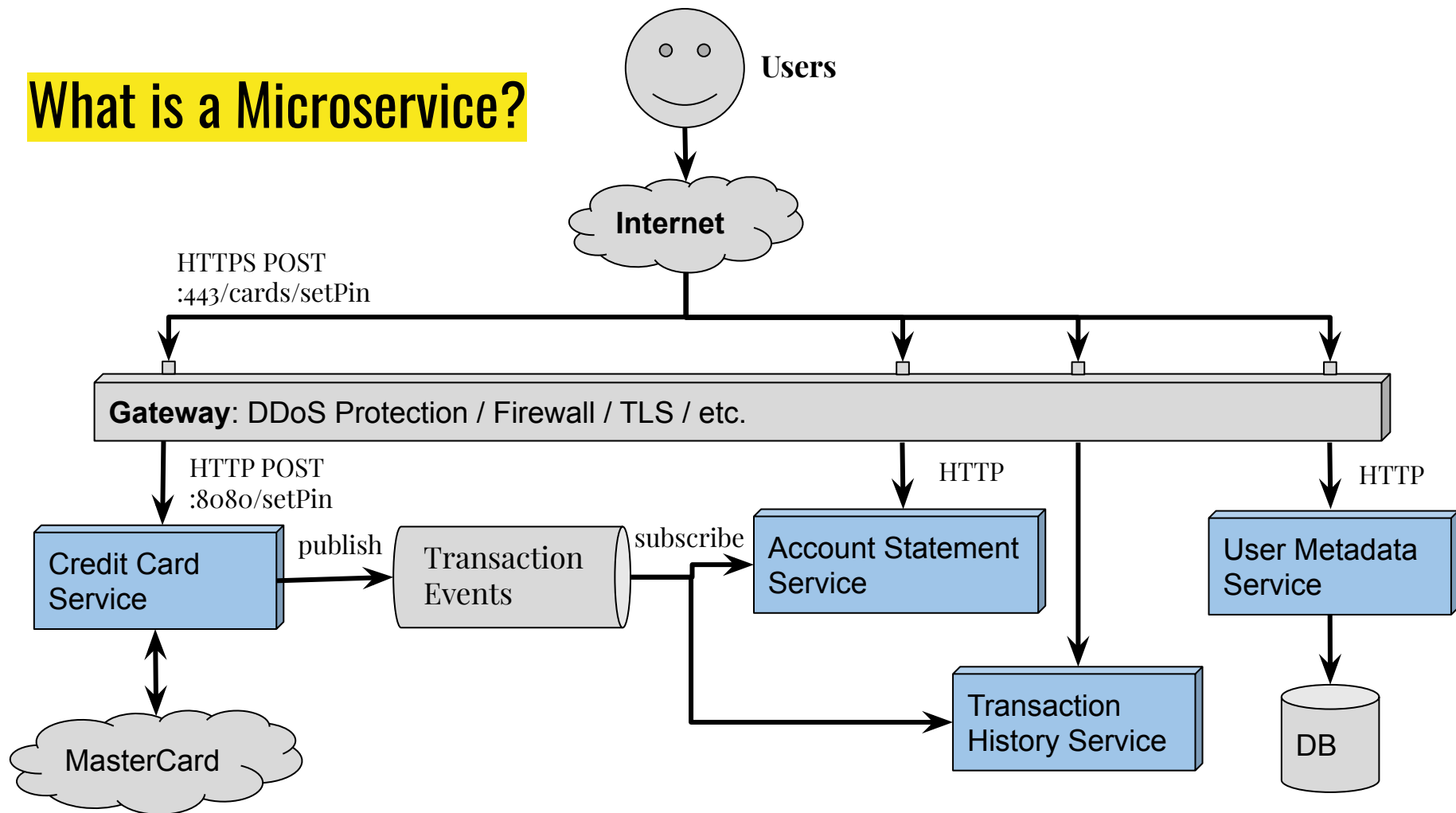# What is a Microservice?

A brief bit of context.

Microservices refer to services that constitute a microservice architecture.

A microservice architecture is a variant of service-oriented architecture, which separates an application into loosely coupled software services.

These services communicate with each other via lightweight protocols:

- Synchronous: REST using HTTP/JSON, gRPC, etc,
- Asynchronous: Apache Kafka, RabbitMQ, AWS SQS, etc.

What is a Microservice?

Users

Internet

HTTPS POST
:443/cards/setPin

Gateway: DDoS Protection / Firewall / TLS / etc.

HTTP POST
:8080/setPin

HTTP

HTTP

Credit Card Service

publish

Transaction Events

subscribe

Account Statement Service

User Metadata Service

MasterCard

Transaction History Service

DB

Why bother?

- Allows different services to use different technologies
- De-couple work so that teams of people can work independently
- Define clear contracts that help system development and testing
- Isolate changes related to scaling/performance to where they are needed
- Make it easier to deploy more frequent and smaller changes

Such architectures are now the norm in many industries.

Microservices and Their Costs

Beyond the interfaces, scaling and deployment are important factors.
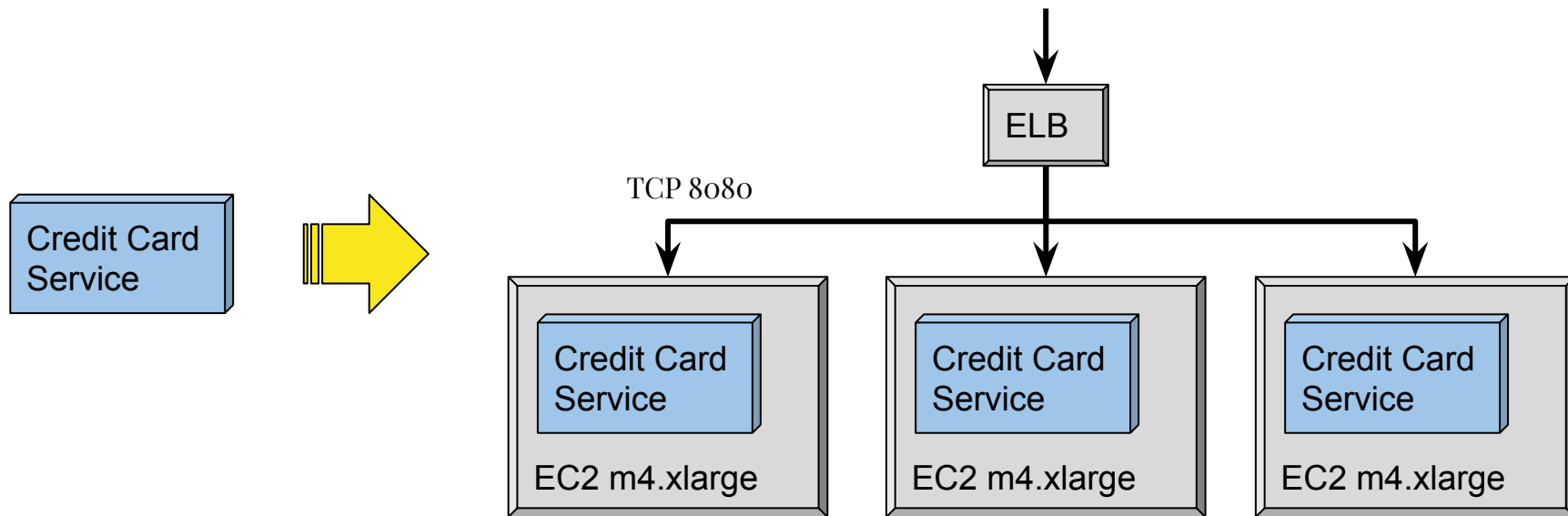
*Each m4.xlarge costs $0.20 per hour (on-demand), i.e. $144.00 per month.

Cutting Costs with Container Orchestration

Low efficiency deployments were enough for first-to-market services in the 2000's and 2010's. In the 2020's, <u>competition and profitability are increasingly important</u>.

Container orchestration systems like <u>Kubernetes</u>, <u>AWS ECS</u>, or <u>OpenShift</u> help configure and deploy multiple services per computer.
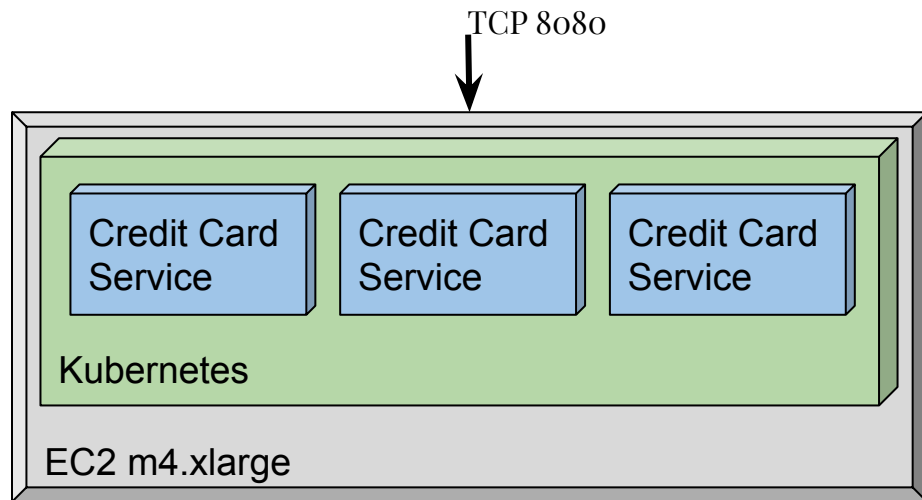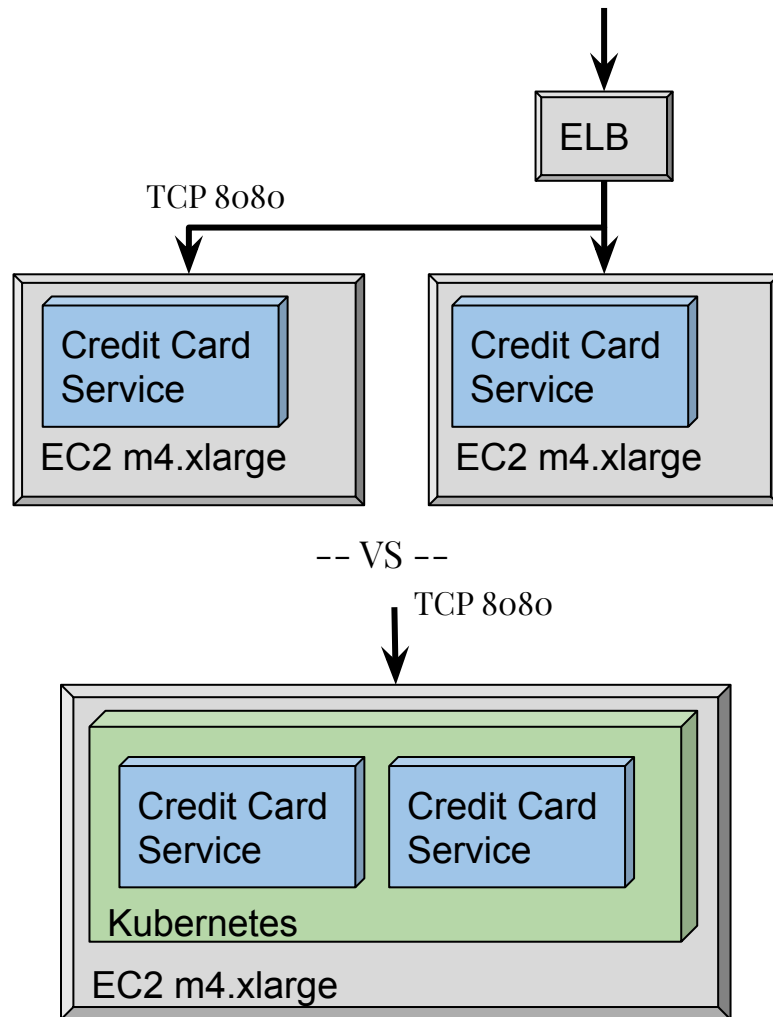
Microservices and Their Costs

Many types of costs per server:

- Hourly Pricing for CPU/RAM
- Hourly Disk Space per GB
- Network Bandwidth per GB
- Public IP Addresses
- Load Balancer Connections

Container orchestration can help with costs depending on pricing.

ELB

TCP 8080

Credit Card Service

EC2 m4.xlarge

Credit Card Service

EC2 m4.xlarge

-- VS --

TCP 8080

Credit Card Service

Credit Card Service

Kubernetes

EC2 m4.xlarge

Cutting Costs with Container Orchestration

Savings exist for multiple mostly-idle services that need <1 CPU.

However, <u>savings for fully loaded servers is limited</u>...

E.g. 2 smaller servers cost as much as 1 bigger server.

Even so, there are savings for:
- Load Balancers
- Public IPs
- Low-use services

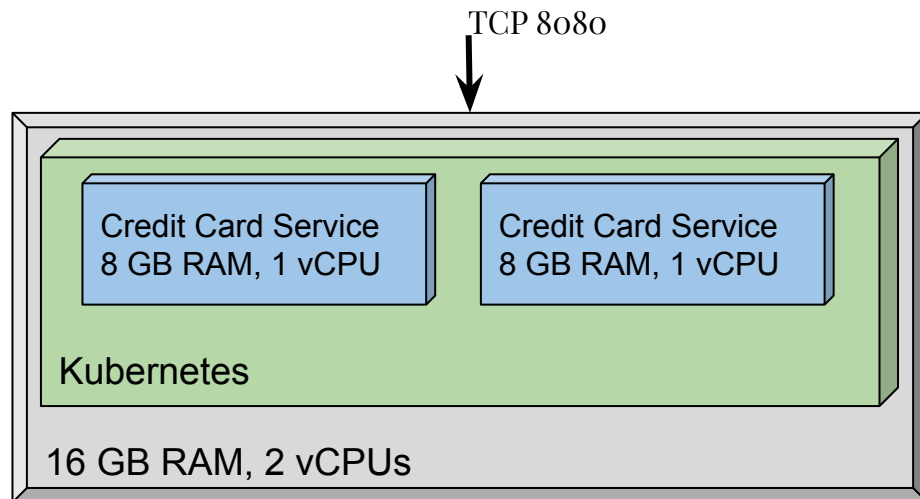| Instance name ▲	On-Demand hourly rate ▽	vCPU ▽	Memory ▽
c7g.medium	$0.0361	1	2 GiB
c7g.large	$0.0723	2	4 GiB
c7g.xlarge	$0.1445	4	8 GiB
c7g.2xlarge	$0.289	8	16 GiB

A Place for System-Level Languages

Get what you pay for, Pay for what you get

Cloud-computing platforms like AWS have become quite efficient.

Two 1-CPU computers cost roughly as much as one 2-CPU computer.

For <u>big companies</u> with 10–100K USD costs per month, <u>resources needs are important</u>.
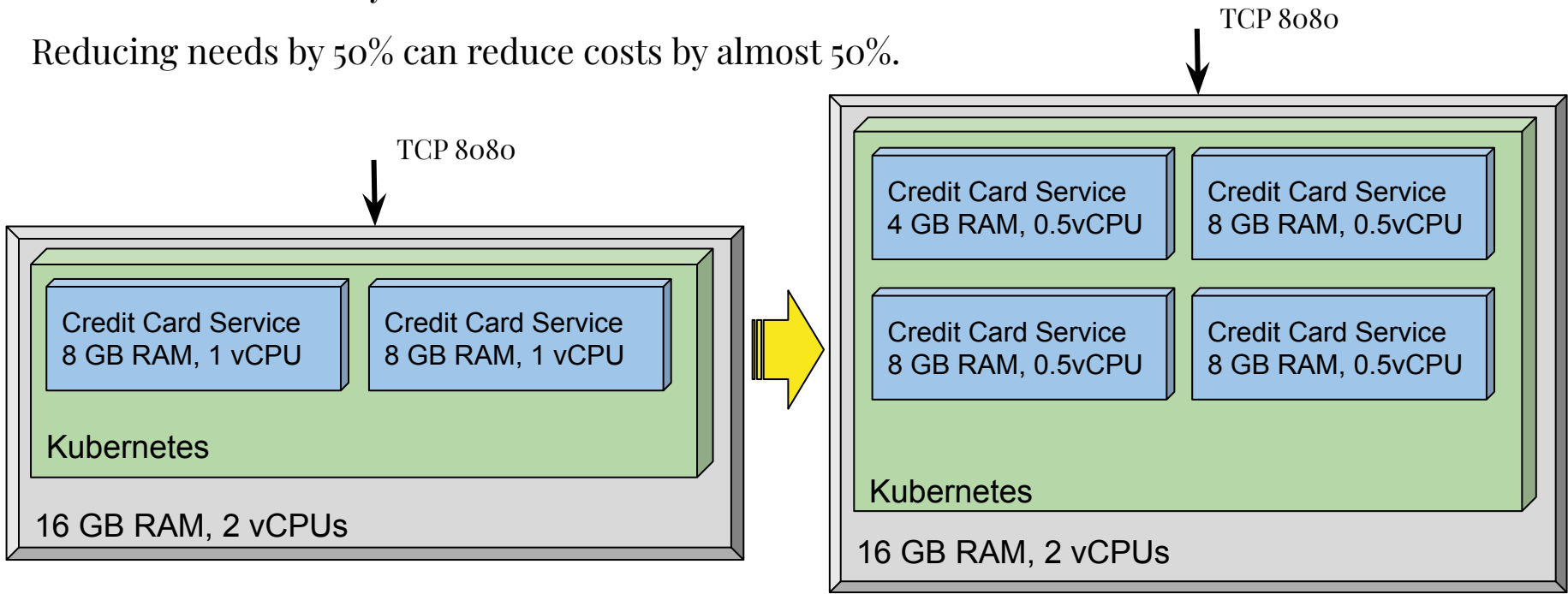
For <u>small companies</u> with a small budget, <u>resources needs are important</u>.

TCP 8080

| Credit Card Service 8 GB RAM, 1 vCPU | Credit Card Service 8 GB RAM, 1 vCPU |

Kubernetes

16 GB RAM, 2 vCPUs

Waste Not, Want Not

The most effective way to cut costs is to need less.

Reducing needs by 50% can reduce costs by almost 50%.

TCP 8080

Credit Card Service
8 GB RAM, 1 vCPU

Credit Card Service
8 GB RAM, 1 vCPU

Kubernetes

16 GB RAM, 2 vCPUs

TCP 8080

Credit Card Service
4 GB RAM, 0.5vCPU

Credit Card Service
8 GB RAM, 0.5vCPU

Credit Card Service
8 GB RAM, 0.5vCPU

Credit Card Service
8 GB RAM, 0.5vCPU

Kubernetes

16 GB RAM, 2 vCPUs

Lower Resource Usage with System-Level Languages

Higher level programming languages are easy to work with, but aren't free:

- **Python**, **JavaScript**, **Ruby**: An <u>interpreter</u> must exist to execute the code, reducing performance and increasing memory usage.
- **Java, Scala, C#, Erlang**: Code is compiled for use by a <u>virtual machine</u>, which is more efficient than an interpreter, but still not free.

System–Level languages compile directly to machine code with little overhead:

- **D, C++, C, Rust, Go, Ada, Fortran**

D vs. Java

A comparison between a minimal web-server using:

- Java / Spring
  - Binary Size: 16.8 MB
  - Resident Memory: 280   MB
  - Virtual Memory: 10,000   MB
  - Transactions/second: 5,200   trans/sec
- D / Vibe.d
  - Binary Size: 3.8 MB
  - Resident Memory: 11   MB
  - Virtual Memory: 510   MB
  - Transactions/second: 11,000   trans/sec

Sönke Ludwig has done fantastic work, check it out at https://vibed.org

Reasons Not to use System-Level Languages

Typical reasons to not use system-level languages include:

- Higher complexity and development costs.
  (Still true, but powerful libraries reduce this barrier.)
- Lack of libraries for certain domains, e.g. AWS, Kafka, Web, ML, etc.
  (As important as language selection is the libraries for that language.)
- Having to compile separately for multiple platforms.
  (With cloud-infrastructure, this has become far less of an issue.)
- Long compilation times and slower development speeds.
  (Varies considerably from language to language.)

Why D for Microservices?

Advantages of D for Web Services

- **Lower Costs:** D's compiled high performance reduces hardware needs.
- **Quick Training:** D's C-family syntax speeds up onboarding for developers already familiar with C++, JavaScript, Python, Java, etc.
- **Incorporate Libraries**: Because D's multi-paradigm feature-rich approach, libraries from other programming languages can be mechanically converted.
- **Fast Development**: D, like Java, has garbage collection. The latency is acceptable and it simplifies development.
- **Error Detection**: D's integration of unit tests and pre/post-conditions makes it easier to write tests, encouraging better test coverage.

Challenges of Using D for Web Services (compared to Java)

- **Lack of IDE Integration:** Java has powerful IDE support ([IntelliJ IDEA](#)) and powerful debugging tools.
  (D [debuggers](#) lack features like code execution.)
- **Lack of Profiling**/**Memory Tools:** Java has powerful inspection tools like [VisualVM](#). (D [profiling](#) is complex, tools often not maintained.)
- **Poor Library Coverage:** Frequently used libraries for Web Services are missing in D.
  - AWS SQS, SNS, Apache Kafka
  - Database ORM (e.g. PostgreSQL)
  - Rule Engines (e.g. [Drools](#))
  - Security (e.g. [Vault](#))

Overcoming Challenges: IDE Integration

IDE Integration:

- **VSCode**: General-use IDE.
  - **Extensions for D**: The user WebFreak has done a good job adding D support.
  - **Syntax Highlighting**: Looks great, responsive, welcoming.
  - **gdb**, **lldb**: Partial support
    - most expressions beyond primitives cannot be evaluated.
    - class members not displayed when debugging class function
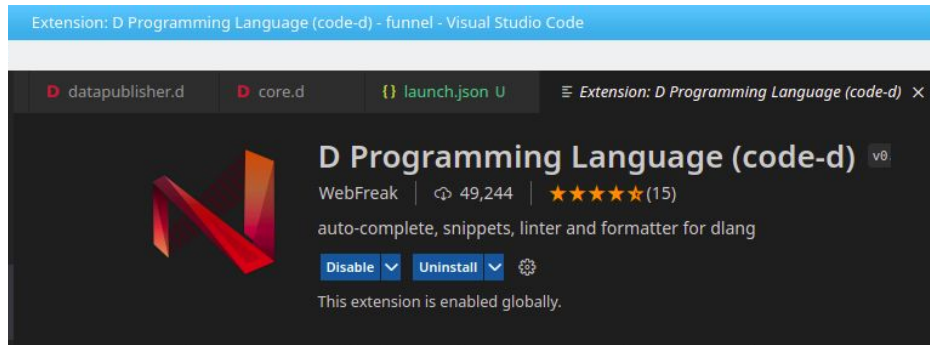
File   Edit   Selection   View   Go   Run   Terminal   Help

RUN AND DEB...   ▷ Debug   ⌄   ⚙   ···

D app.d   ⠿ ▷ ⟳ ↧ ↥ ↺ ◁ ↻ ◻   {} launch.json U   ⊟ ···

VARIABLES

mouth > source > funnel > service > D datapublisher.d > ⠂ DataPublisher > ⬡ startTimer

> __capture: <args>

> __r3235: <unknown>

__key3236: 0

⌄ name: Object@*0x7ffff7547d40

__exception_object: Could not expand v...

WATCH

```
38         setTimer(
39             750.msecs,
40             {
41                 foreach (Name name; _schemaRegistry.getSchemaNames()) {
42                     // The data is now gone, make sure it gets sent.
43                     ubyte[] dataBytes;
44                     ushort count;
45                     _dataService.getAndClearData(name, count, dataBytes);
46                     if (dataBytes.length == 0)
47                         continue;
48                     auto bundle = new GenericDataBundle(name, count, dataBytes);
49                     logDebug("Writing to stem: GenericBundle[name=%s, count=%d, dataBytes
50                         bundle.name, bundle.count, bundle.data.to!string);
```

CALL STACK

⌄ 1:Thread 0x7ffff7639600...   PAUSED ON STEP

funnel.service.datapublisher.DataPublishe

_D4vibe4coreQf8setTimerFNbSQu4time8Durati

_D4vibe4coreQf11createTimerFNbNfDFNbNfZv

_D4vibe4core4task12TaskFuncInfo__T3setTDF

BREAKPOINTS

● ☑ app.d   mouth/source   42

● ☑ datapublisher.d   mouth/source/fu...   38

● ☑ datapublisher.d   mouth/source/fu...   43

● ☑ datapublisher.d   mouth/source/fu...   45

PROBLEMS   OUTPUT   **DEBUG CONSOLE**   TERMINAL   Filter (e.g. text, !exclude)   ≡ ⌄ ∧ ✕

```
43                     ubyte[] dataBytes;
② [Switching to thread 10 (Thread 0x7fffdeffd700 (LWP 23817))](running)

Thread 1 "funnel-mouth" hit Breakpoint 4, funnel.service.datapublisher.DataPublisher.startTimer().__lambda6() (__ca
pture=0x7ffff7563060) at source/funnel/service/datapublisher.d:45
45                     _dataService.getAndClearData(name, count, dataBytes);
[Switching to thread 10 (Thread 0x7fffdeffd700 (LWP 23817))](running)
p _dataService
No symbol "_dataService" in current context.
No symbol "_dataService" in current context. (from interpreter-exec --thread 1 --frame 0 console "p _dataService")
```
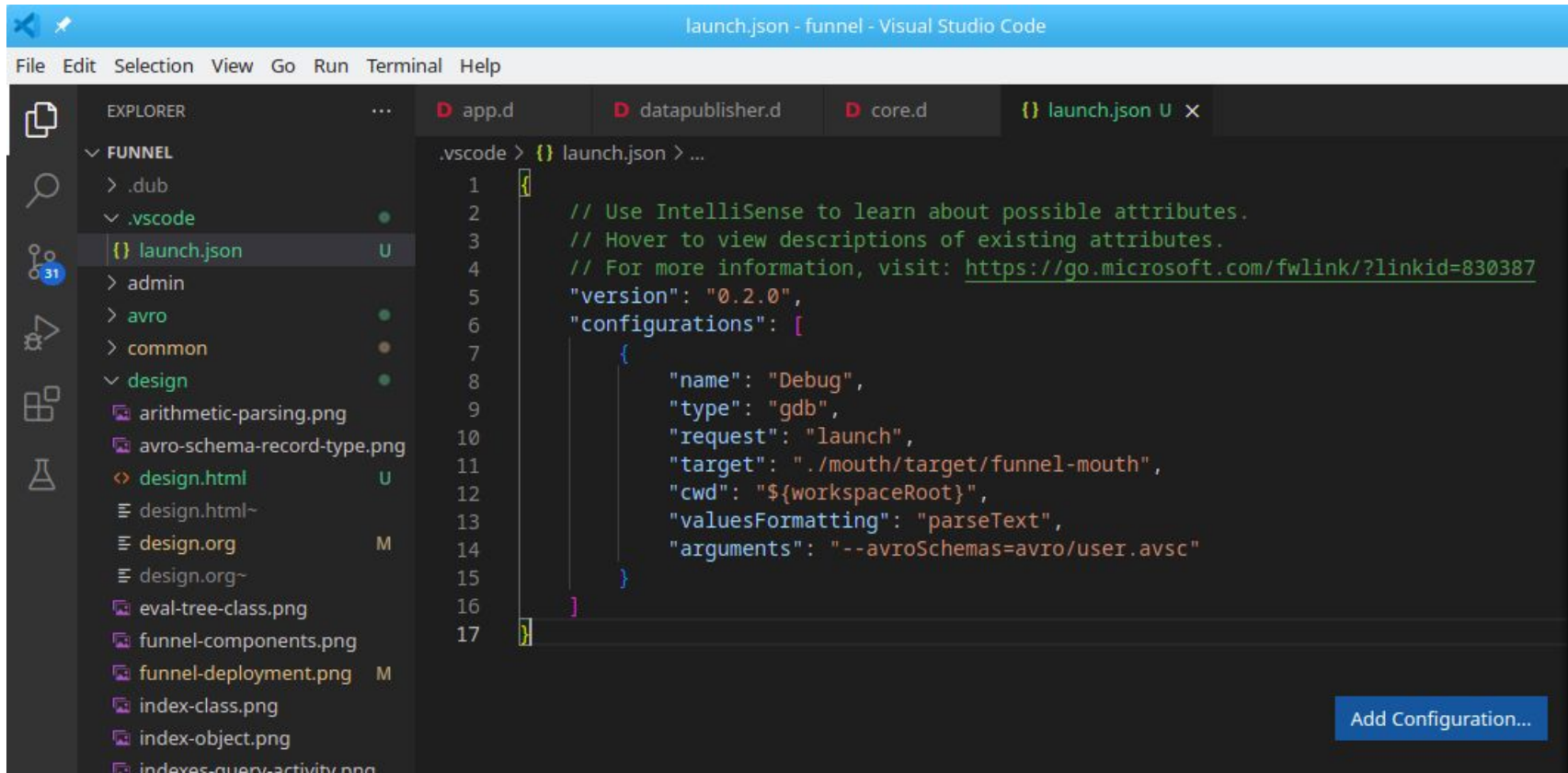
>

⤴ master*   ↻   ⊗ 0 ⚠ 0   Debug (funnel)   application   debug   dmd   Ln 46, Col 1   Spaces: 2   UTF-8   LF   D
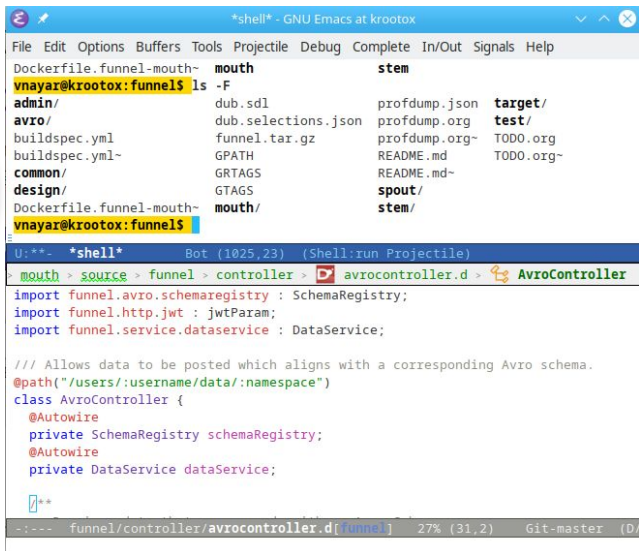
Debug and other configurations must be established by hand without auto–detection.


```json
{
    // Use IntelliSense to learn about possible attributes.
    // Hover to view descriptions of existing attributes.
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
    "version": "0.2.0",
    "configurations": [
        {
            "name": "Debug",
            "type": "gdb",
            "request": "launch",
            "target": "./mouth/target/funnel-mouth",
            "cwd": "${workspaceRoot}",
            "valuesFormatting": "parseText",
            "arguments": "--avroSchemas=avro/user.avsc"
        }
    ]
}
```

Despite existing configurations in dub, D IDE development is less "plug-and-play" than Java/Maven with IntelliJ or Eclipse.

Old Fashioned Tools Still Work

If speed and low resource usage are more important than good looks, classic tools still continue to work:

File   Edit   Options   Buffers   Tools   Projectile   Gud   Complete   In/Out   Signals   Help

```
45                    _dataService.getAndClearData(name, count, dataBytes);
(gdb) n
[Switching to thread 7 (Thread 0x7fffdffff700 (LWP 28178))](running)
46                    if (dataBytes.length == 0)
(gdb) p name
$1 = (avro.name.Name *) 0x7ffff7547d40
(gdb) p name.namespace
$2 = "funnel"
(gdb)
```

Locals Registers

```
_Array_*                                            __r3235     <complex data type>
uint long long                                      __key3236   0
avro.name.Name *                                    name        0x7ffff7547d40
_Array_unsigned char                                dataBytes   <complex data type>
ushort                                              count       0
funnel.avro.genericdatabundle.GenericDataBundle * volatile bundle  0x7ffff75602c0
_Array_unsigned char                                bundleBytes <complex data type>
int                                                 bytesSent   1442607376
```

U:**- *gud-funnel-mouth*   Bot (115,6)   (Debugger:run [end-stepping-range] +1 Projectile)   U:%*- *locals of funnel-mouth* All (1,0)   (Locals: funnel.service.datapublisher.DataPublisher.startTimer():.__la

mouth > source > funnel > service > D▸ datapublisher.d > DataPublisher > startTimer

```
      enforce(nn_connect(sock, url) >= 0);

          setTimer(
              750.msecs,
              {
                  foreach (Name name; _schemaRegistry.getSchemaNames()) {
                      // The data is now gone, make sure it gets sent.
                      ubyte[] dataBytes;
                      ushort count;
                      _dataService.getAndClearData(name, count, dataBytes);
                      if (dataBytes.length == 0)
                          continue;
                      auto bundle = new GenericDataBundle(name, count, dataBytes);
                      logDebug("Writing to stem: GenericBundle[name=%s, count=%d, dataBytes=%s]",
                          bundle.name, bundle.count, bundle.data.to!string);
                      ubyte[] bundleBytes = bundle.toBytes();
                      int bytesSent = nn_send(sock, bundleBytes.ptr, bundleBytes.length, 0);
                      if (bytesSent < dataBytes.length) {
                          // TODO: Do something if the bytes could not be sent.
```

```
source/app.d:38 _Dmain [0x555555a2785a]
JWT tokens are only validated if the environment variable JWT_PUBLIC_KEY or JWT_PUBLIC_KEY_URL
is set.
[main(----) INF] Parsing Avro Schema: ../avro/user.avsc
[main(----) INF] Parsing Avro Schema: ../avro/user.avsc
[main(----) INF] Connecting to stemSubUrl: tcp://127.0.0.1:5000
[main(----) INF] Parsing Avro Schema: ../avro/user.avsc
[main(----) INF] Please open http://127.0.0.1:8080/ in your browser.
[vibe-2(C+MH) INF] Parsing Avro Schema: ../avro/user.avsc
[vibe-2(C+MH) INF] Listening for requests on http://[::]:8080/
[vibe-2(C+MH) INF] Listening for requests on http://0.0.0.0:8080/
[vibe-0(W5nj) INF] Listening for requests on http://[::]:8080/
[vibe-0(W5nj) INF] Listening for requests on http://0.0.0.0:8080/
[vibe-4(Gwlc) INF] Listening for requests on http://[::]:8080/
[vibe-4(Gwlc) INF] Listening for requests on http://0.0.0.0:8080/
[vibe-1(gGuM) INF] Listening for requests on http://[::]:8080/
[vibe-1(gGuM) INF] Listening for requests on http://0.0.0.0:8080/
[vibe-5(Hf39) INF] Listening for requests on http://[::]:8080/
[vibe-5(Hf39) INF] Listening for requests on http://0.0.0.0:8080/
[vibe-6(fW1b) INF] Listening for requests on http://[::]:8080/
```

-:--- funnel/service/datapublisher.d [funnel]   53% (46,0)   Git-master   (D// +1 Lens LSP[serve-d:9121] Projectile[fun   U:%*- *input/output of funnel-mouth* 75% (31,65) (Inferior I/O:run +1 Projectile)

```
0 in funnel.service.datapublisher.DataPublisher.startTimer().__lambda6() of source/funnel/serv
1 in _D4vibe4coreQf8setTimerFNbSQu4time8DurationDFZvbZ9__lambda4MFNbNeZv of ../../../.dub/packa
2 in _D4vibe4coreQf11createTimerFNbNfDFNbNfZvZ1C6opCallMFNbNfSQCdQCbQCe5TimerZ9__lambda2MFNbNf(
3 in _D4vibe4core4task12TaskFuncInfo__T3setTimerFNbNfSQBsQBqQBt5TimerZvTQtZQBhMFKQBiKQBgZ12callDe
4 in vibe.core.task.TaskFuncInfo.call() of ../../../.dub/packages/vibe-core-1.22.4/vibe-core/so
5 in vibe.core.task.TaskFiber.run() of ../../../.dub/packages/vibe-core-1.22.4/vibe-core/source
6 in core.thread.context.Callable.opCall()
7 in fiber_entryPoint
8 in ??
```

Breakpoints Threads

```
Num Type          Disp Enb Addr               Hits What
1   breakpoint keep y   <MULTIPLE>          10   in unknown
2   breakpoint keep y   <MULTIPLE>          10   in unknown
3   breakpoint keep y   0x0000555555a27949 9    in app.main().__lambda3(shared(poodinis.contai
ner.DependencyContainer), ushort) of source/app.d:47
4   breakpoint keep y   0x0000555555a81331 2    in funnel.service.datapublisher.DataPublisher.
startTimer().__lambda6() of source/funnel/service/datapublisher.d:45
```

U:%*- *stack frames of funnel-mouth*   All (1,0)   (Frames [thread 1] +1 Projectile)   U:%*- *breakpoints of funnel-mouth* All (1,0)   (Breakpoints +1 Projectile)

Overcoming Challenges: Profiling Tools

Profiling Tools: Valgrind

- Running through Valgrind (D Wiki: <u>Other Dev Tools</u>)
  - Run program through Valgrind, which has a smaller performance impact.
    ```
    $ valgrind --tool=callgrind --collect-jumps=yes
    --callgrind-out-file=callgrind_out
    ./mouth/target/funnel-mouth -- --avroSchemas=avro/user.avsc
    ```
  - Demangle D symbols to make them human readable:
    ```
    $ ddemangle callgrind_out > callgrind_out.demangle
    ```
  - View results using kcachegrind.
    ```
    $ kcachegrind callgrind_out.demangle
    ```

File   View   Go   Settings   Help

Open...   ◁ Back ▾   ▷ Forward ▾   ⌃ Up ▾   ⟋ Relative   ↻ Cycle Detection   ✛ Relative to Parent   ⟨⟩ Shorten Templates | Instruction Fetch   ▾ |

Flat Profile   ◇ ✕

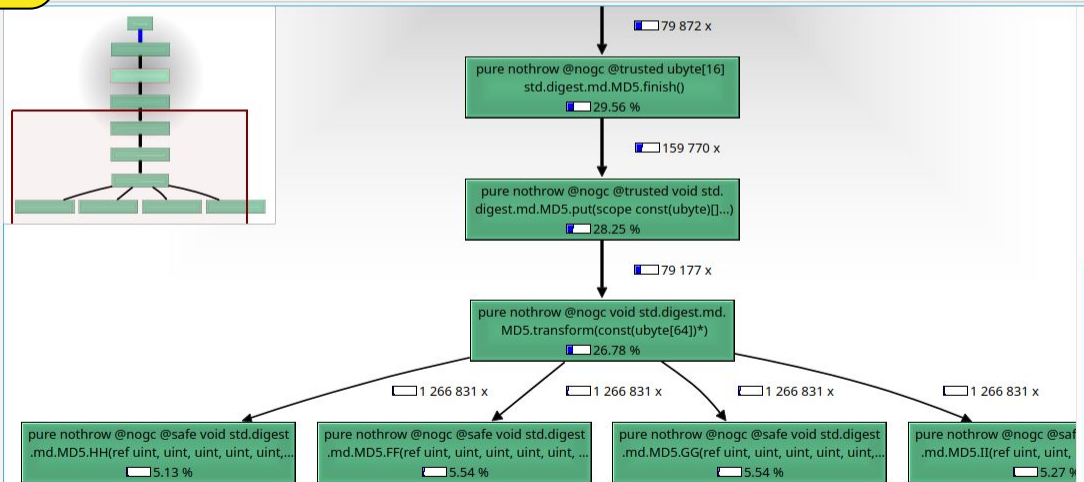Search: | Search Query | | ELF Object   ▾ |

pure nothrow @nogc @safe ubyte[16] std.digest.md.md5Of(void[]).md5Of(void[])

| Types | Callers | All Callers | Callee Map | Source Code |

Self	ELF Object
94.24	■ funnel-mouth
3.40	■ libgcc_s.so.1
1.50	■ libc-2.31.so
0.27	■ ld-2.31.so
0.24	■ libcrypto.so.1.1

Event Type	Incl.	Self	Short	Formula
Instruction Fetch	■ 30.33	0.09	Ir	
Cycle Estimation	■ 30.33	0.09	CEst =	Ir

Incl.	Self	Called	Function	
■ 35.76	0.38	79 872	const pure nothrow @nogc ulong[2] vil...	
■ 34.29	0.45	92 673	pure nothrow @nogc @trusted ubyte[1...	
■ 33.07	1.73	278 036	pure nothrow @nogc @trusted void st...	
■ 31.34	6.21	92 673	pure nothrow @nogc void std.digest.md.MD5.f...(const(ubyte[64])*)	funnel-mouth
■ 30.33	0.09	79 872	pure nothrow @nogc @safe ubyte[16] std.digest.md.md5Of(void[]).md5Of...	funnel-mouth: r
■ 30.24	0.30	79 872	pure nothrow @nogc @safe ubyte[16] std.digest.digestl(std.digest.md.MD...	funnel-mouth: p
┃ 9.34	1.65	8	@trusted void vibe.http.router.MatchGraphBuilder.disambiguate() <cycle 4>	funnel-mouth: r
	6.62	0.00	8	nothrow ulong core.internal.gc.impl.conservative.gc.Gcx.fullcollect(bool, b...	funnel-mouth
	6.61	0.00	8	nothrow void core.internal.gc.impl.conservative.gc.Gcx.markParallel(bool)	funnel-mouth
	6.48	4.42	1 482 768	pure nothrow @nogc @safe void std.digest.md.MD5.FF(ref uint, uint, uint, ...	funnel-mouth
	6.48	4.42	1 482 768	pure nothrow @nogc @safe void std.digest.md.MD5.GG(ref uint, uint, uint,...	funnel-mouth
	6.21	0.40	15 692	pure @safe immutable(char)[] std.json.parseJSON!(immutable(char)[]).pars...	funnel-mouth: j
	6.16	4.42	1 482 768	pure nothrow @nogc @safe void std.digest.md.MD5.II(ref uint, uint, uint, u...	funnel-mouth
	6.11	0.00	8	nothrow void core.internal.gc.impl.conservative.gc.Gcx.collectAllRoots(bool)	funnel-mouth
	6.09	6.09	2 358	nothrow scope void core.internal.gc.impl.conservative.gc.Gcx.collectRoots(...	funnel-mouth
	6.01	4.42	1 482 768	pure nothrow @nogc @safe void std.digest.md.MD5.HH(ref uint, uint, uint,...	funnel-mouth
	5.88	0.00	8	nothrow int core.internal.container.treap.Treap!(core.gc.gcinterface.Range)...	funnel-mouth
	5.88	0.00	8	nothrow int core.internal.container.treap.Treap!(core.gc.gcinterface.Range)...	funnel-mouth
	5.87	0.00	2 043	nothrow int core.internal.container.treap.Treap!(core.gc.gcinterface.Range)...	funnel-mouth
	5.87	0.00	2 043	nothrow int core.internal.gc.impl.conservative.gc.Gcx.collectAllRoots(bool)....	funnel-mouth
	4.87	0.01	12 784	pure nothrow @nogc @safe ubyte[16] std.digest.md.md5Of(const(ubyte)[])...	funnel-mouth: r
	4.85	0.05	12 784	pure nothrow @nogc @safe ubyte[16] std.digest.digestl(std.digest.md.MD...	funnel-mouth: p
	4.60	0.72	235 096	pure @safe char std.json.parseJSON!(immutable(char)[]).parseJSON(immut...	funnel-mouth: j
	4.16	0.01	2 043	nothrow int core.internal.container.treap.Treap!(core.gc.gcinterface.Range)...	funnel-mouth
	3.79	3.79	5 931 072	pure nothrow @nogc @safe uint core.bitop.rol!(uint).rol(const(uint), const(...	funnel-mouth

> **Looks like validating JWT tokens is costly!**

(call graph)

79 872 x

┌──────────────────────────────┐
│ pure nothrow @nogc @trusted ubyte[16] │
│ std.digest.md.MD5.finish()   │
│   ▫ 29.56 %   │
└──────────────────────────────┘

159 770 x

┌──────────────────────────────┐
│ pure nothrow @nogc @trusted void std. │
│ digest.md.MD5.put(scope const(ubyte)[]...) │
│   ▫ 28.25 %   │
└──────────────────────────────┘

79 177 x

┌──────────────────────────────┐
│ pure nothrow @nogc void std.digest.md. │
│ MD5.transform(const(ubyte[64])*)   │
│   ▫ 26.78 %   │
└──────────────────────────────┘

1 266 831 x   1 266 831 x   1 266 831 x   1 266 831 x

┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐ ┌─────────────────┐
│ pure nothrow @nogc @safe void std.digest │ │ pure nothrow @nogc @safe void std.digest │ │ pure nothrow @nogc @safe void std.digest │ │ pure nothrow @nogc @saf │
│ .md.MD5.HH(ref uint, uint, uint, uint, uint,... │ │ .md.MD5.FF(ref uint, uint, uint, uint, ... │ │ .md.MD5.GG(ref uint, uint, uint, uint, uint, ... │ │ .md.MD5.II(ref uint, uint, │
│   ▫ 5.13 %   │ │   ▫ 5.54 %   │ │   ▫ 5.54 %   │ │   ▫ 5.27 %   │
└─────────────────┘ └─────────────────┘ └─────────────────┘ └─────────────────┘

| Parts | Callees | Call Graph | All Callees | Caller Map | Machine Code |

Overcoming Challenges: Memory Tools

Memory Profiling: Compiler option "-profile=gc"

- May also be enabled via dub through "dub build --build=profile-gc"
- Memory allocations are output to "profilegc.log"
- Output is reasonably formatted (albeit with long names), e.g.:

```
bytes allocated, allocations, type, function, file:line
     4096000                 500     ubyte[] vibe.internal.array.BatchBuffer!(ubyte,
0LU).BatchBuffer.capacity
../../.dub/packages/vibe-core-1.22.4/vibe-core/source/vibe/internal/array.d:580
      662688                 20709     std.array.Appender!(immutable(char)[]).Appender.Data
std.array.Appender!string.Appender.this /usr/include/dmd/phobos/std/array.d:3330
      304000                 1000     ubyte[] std.base64.Base64Impl!('-', '_',
'\x00').decode!string.decode /usr/include/dmd/phobos/std/base64.d:1339
```

Overcoming Challenges: Memory Tools

Available tools still have gaps:
- No visual aid for tracking memory allocations.
- No ability to do memory sampling

Languages like Java have such tools available ==>
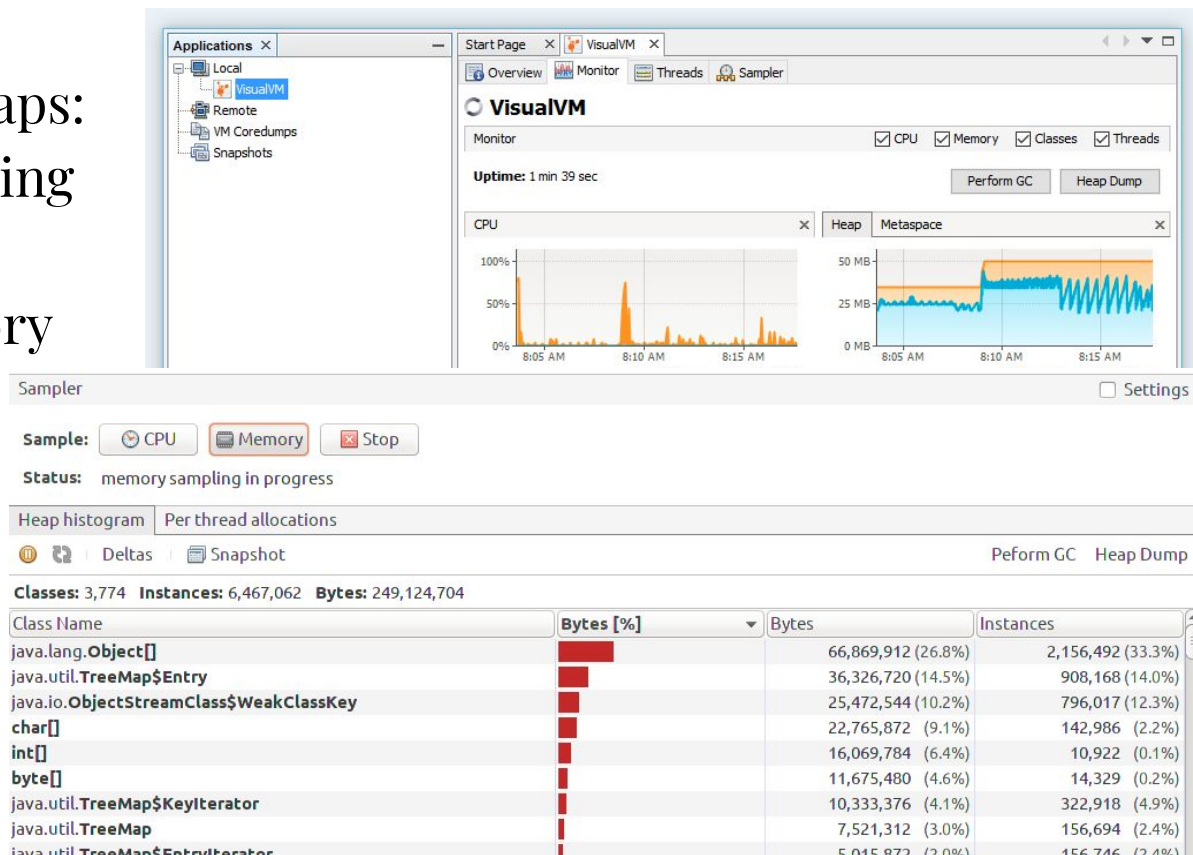
Overcoming Challenges: Profiling Tools to Avoid (for now)

Profiling Tools: dmd -profile

- Theoretically this is built in: Compile using "-profile"
  - Compile using the "-profile" switch or the dub build profile, e.g.
    `$ dub build --build=profile`
- Difficult to use in practice.
  - Compilation frequently fails, e.g.
    `/usr/include/dmd/phobos/std/concurrency.d(2160,17): Warning: statement is not reachable`
  - Compilation using "-profile" produces hard to read results.
  - Performance is impacted considerably.
  - The output format is undocumented and contains no labels.

Overcoming Challenges: Library Coverage

Library Coverage: Common Web Service libraries are missing in D...
...but D is an amazing language for importing libraries.

D's multi-paradigm approach makes it extremely compatible with most other programming languages.

	D	Python	C++	Java	Go
Garbage Collection	✔	✔	✖	✔	✔
Pointer Manipulation	✔	✖	✔	✖	✖
Object Oriented	✔	✔	✔	✔	✖
Generic Programming	✔	✔	✔	✔	✔
Fast Prototyping	✔	✔	✖	~	✔

Porting Libraries to D

D's familiar syntax makes it easy to quickly convert libraries from other languages with much less effort than usual.

For converting large libraries, a common strategy is:

1. Understand the chain of dependencies in the library, build outward starting from the deepest dependencies (typically utils).
2. Convert the library as literally and mechanically as possible, alterations can have cascading impacts which are hard to track.
3. Be sure to migrate tests and validate work as it moves along.
4. Refactor a single feature at a time.

Porting Libraries to D

Typical range algorithms don't always work when converting libraries. Iterator-capable containers are often needed to port algorithms.

```
// Position the iterator at the first cell that overlaps or follows
// "target", i.e. such that range_max() >= target.range_min().
void SeekTo(const RangeIterator& target) {
  it_.Seek(target.range_min());
  // If the current cell does not overlap "target", it is possible that the
  // previous cell is the one we are looking for.  This can only happen when
  // the previous cell contains "target" but has a smaller S2CellId.
  if (it_.done() || it_.id().range_min() > target.range_max()) {
    if (it_.Prev() && it_.id().range_max() < target.id()) it_.Next();
  }
  Refresh();
}
```

Porting Libraries to D

Another example algorithm that requires iterators to be properly converted. In D, such containers are simply not part of the standard library, but can be implemented, e.g. btree.d and btree_map.d.

```d
// TODO(ericv): Use a single iterator (iter_) below and save position
// information using pair<S2CellId, const S2ShapeIndexCell*> type.
auto next = new S2ShapeIndex.Iterator(
    _index, S2ShapeIndex.InitialPosition.BEGIN);
auto last = new S2ShapeIndex.Iterator(
    _index, S2ShapeIndex.InitialPosition.END);
last.prev();
if (next.id() != last.id()) {
  // The index has at least two cells.  Choose a level such that the entire
  // index can be spanned with at most 6 cells (if the index spans multiple
  // faces) or 4 cells (it the index spans a single face).
  int level = next.id().getCommonAncestorLevel(last.id()) + 1;
  // Visit each potential top-level cell except the last (handled below).
  S2CellId last_id = last.id().parent(level);
```

Putting it Into Practice

A Practical Problem

What do the following industries have in common?

- Ride Hailing
- Micro Mobility
- IoT and Sensor Monitoring
- Military/Security Situational Awareness

A Practical Problem

What do the following industries have in common?

- Ride Hailing
- Micro Mobility
- IoT and Sensor Monitoring
- Military/Security Situational Awareness

They all have very high volumes of data updates and queries in real-time.

By the Numbers

- 1,000 Taxis per City
- 50 Cities
- 1 position/status update per 10 seconds

5,000 updates per second!

- 10M people per city
- 5% of users look at the app per day
- each user session makes 5 queries

1,500 queries per second!

Numbers get big, fast.

Need for Speed

Customers need flexible queries... an RDS would fit.

Write performance needs to be extreme... a data-cache would fit.

Traditional solutions get VERY expensive as the system grows and grows, starting from a simple RDS, to multiple read-replicas, to sharding, to cache layers in Redis, etc.
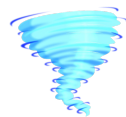
🌪️ Funnel-Labs.io was created to tackle this challenge with a ready-made solution.

A Brief Look

A simple JavaScript example client.

http://demo-query.funnel-labs.io/

Data format with coordinates to query.

Query results shown visually.

What does it look like to a user?

```
{
 "namespace": "com.example",
 "name": "Staff",
 "type": "record",
 "fields": [
  {
    "name": "loc",
    "type": {
     "type": "record",
     "name": "Location",
     "fields": [
       {"name": "latitude", "type": "double"},
       {"name": "longitude", "type": "double"}
     ]
   },
   "idx_type": "spatial"
  },
  ...
```

1. Authenticate and get a JWT token.

2. Upload a schema.

3. Upload your data as desired.

4. Execute queries on the data.

```
HTTP POST
https://auth.funnel-labs.io/auth/realms/fun
nel/protocol/openid-connect/token

client_id=service
&grant_type=password
&username=ExampleInc
&password=abcd1234
```

```
$ curl
https://api.funnel-labs.io/users/ExampleInc/data/com
.example/Staff \
 -X POST -d '
 {"id": 3,
  "job": "WAITER",
  "loc": {"latitude": 1.0, "longitude": 2.0},
  "rating": 1.3,
  "name": "Bob"}' \
 -H 'Content-Type: application/json' \
 -H "Authorization: Bearer $ACCESS_TOKEN"

OK
```
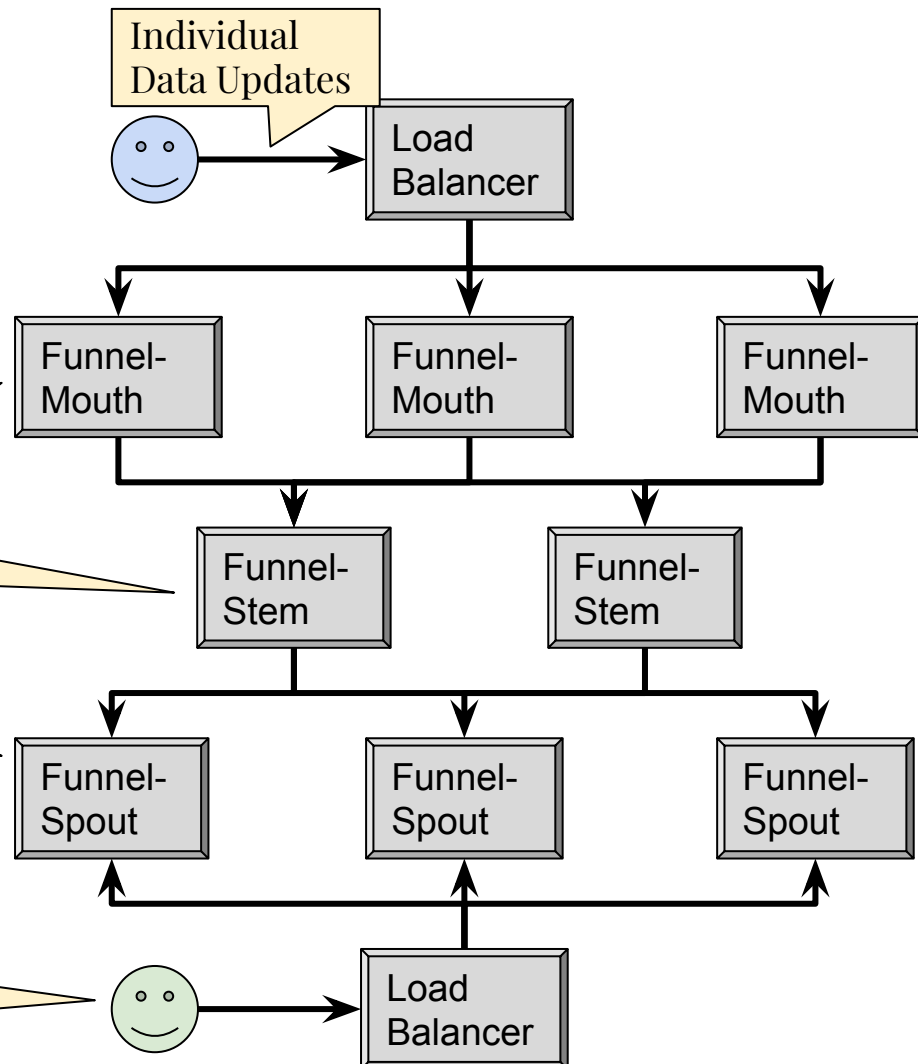
```
loc WITHIN 2km of (13.34, 43.21)

rating > 2.5 AND type = ECONOMY
```

Basic Structure of Funnel System

The Funnel System was built to solve this problem as cheaply as possible.

Individual Data Updates

Load Balancer

Data collected into batches to reduce network load.

Funnel-Mouth

Funnel-Mouth

Funnel-Mouth

Data replicated to spouts using pub/sub in batches.

Funnel-Stem

Funnel-Stem

Data indexed in RAM for queries.

Funnel-Spout

Funnel-Spout

Funnel-Spout

Arbitrary queries pour in.

Load Balancer

Dependencies

- [Vibe.d](): Sönke Ludwig – Web Server and concurrency
- [Pegged](): Philippe Sigaud – Grammar based parser generator

- [S2 Geometry](): Converted to D from C++ as [s2geometry-d]()
  - A team of 1 converted this 70K loc library more completely than Go, Rust, Java, etc.
- [Avro](): Converted to D from Java as [avro-d]()
  - Converted in only a few weeks of work.

Using D, small teams can match the productivity of bigger organizations.

Build Pipeline

Docker Containers: Build Container

The build container is responsible for compiling and uploading to a Docker registry.
Alpine Linux lets you pack the build environment into less than 50MB, therefore faster builds.

```
# Create an Alpine Linux image w/ docker & ldc2.
FROM docker:dind
RUN apk update

# Add tools and dependencies.
RUN apk add --no-cache gcc
RUN apk add --no-cache ldc

# The dub build tool is used to build a D repository.
RUN apk add --no-cache -X
https://dl-cdn.alpinelinux.org/alpine/edge/testing
dub

# Add any C dependencies needed.
RUN apk add --no-cache openssl-dev
RUN apk add --no-cache musl-dev
RUN apk add --no-cache zlib-dev
```

```
# Install AWS CLI tools for use with ECR
RUN apk add --no-cache aws-cli
RUN aws --version

# Create a build user with reduced privileges.
ARG USER=default
ENV HOME /home/$USER

RUN apk add --update sudo
RUN adduser -D $USER \
        && echo "$USER ALL=(ALL) NOPASSWD: ALL" >
/etc/sudoers.d/$USER \
        && chmod 0440 /etc/sudoers.d/$USER

USER $USER
WORKDIR $HOME
```

Build Pipeline

Many options exist, but minimizing engineering cost, server setup, maintenance, etc. can pay off in spades.

A classic [Jenkins](#) server can be set up, but has costs whether in use or not.

Alternatively, cloud-services like [AWS CodePipeline](#) only charge for build-time.

Add a [buildspec.yml](#) file to your project which runs in your build docker container, and you're off to the races.

Dockerized Build/Deploy Environments

When building using cloud-based resources like AWS CodePipeline, they do not know what kind of environment you need to build and run your application.

Generally, you will need 3 different types of docker containers:

- **Build Environment**: Has build tools like a compiler, library headers, and can upload built images to Docker.
- **Runtime Base**: A common environment for running D programs, with D runtime libraries.
- **Project Runtime**: The container for a specific project, with the compiled program and any additional files.

Build Pipeline: buildspec.yml

With AWS CodeBuild, the buildspec.yml runs in your build environment, and builds a deployable image.

```yaml
version: 0.2
env:
  variables:
    # Multiple threads is only supported in ldc2.
    DFLAGS: "--threads=4"
phases:
  install:
    commands:
      - nohup /usr/local/bin/dockerd --host=unix:///var/run/docker.sock
--host=tcp://127.0.0.1:2375 &
      - timeout 15 sh -c "until docker info; do echo .; sleep 1; done"
  pre_build:
    commands:
      # Log into docker and the container registry.
      - AWS_ACCOUNT_ID=$(echo $CODEBUILD_BUILD_ARN | cut -d ':' -f5 )
      - COMMIT_HASH=$(echo $CODEBUILD_RESOLVED_SOURCE_VERSION | cut -c
1-7)
      - IMAGE_NAME=funnel-mouth
      - IMAGE_TAG=${COMMIT_HASH:=latest}
      - ECR_REPO=$AWS_ACCOUNT_ID.dkr.ecr.$AWS_DEFAULT_REGION.amazonaws.com
      - aws ecr get-login-password --region $AWS_DEFAULT_REGION |
        docker login --username AWS --password-stdin $ECR_REPO
```

```yaml
  build:
    commands:
      - echo Build started on `date`
      - cd mouth
      - dub build -b release
      - dub test -b release
      - echo Creating docker image...
      - docker build --build-arg IMAGE_BASE=$ECR_REPO/runtime:latest -t
$IMAGE_NAME:$IMAGE_TAG .
      - docker tag $IMAGE_NAME:$IMAGE_TAG $ECR_REPO/$IMAGE_NAME:$IMAGE_TAG
      - docker tag $IMAGE_NAME:$IMAGE_TAG $ECR_REPO/$IMAGE_NAME:latest
  post_build:
    commands:
      - echo Build completed on `date`
      - echo Pushing the Docker image...
      - docker push $ECR_REPO/$IMAGE_NAME:$IMAGE_TAG
      - docker push $ECR_REPO/$IMAGE_NAME:latest
```

Docker Containers: Example Project Container

Projects have their own Dockerfile, defining what files and arguments are needed.

```
# Container definition for the funnel-mouth project.
ARG IMAGE_BASE=runtime:latest
FROM $IMAGE_BASE

## Environment Variables
# LISTEN_PORT - The port where HTTP requests are received.
# STEM_SUB_HOST - The hostname of a mouth-stem server.

ENV LISTEN_PORT=8080
ENV STEM_SUB_HOST=127.0.0.1

RUN mkdir service

WORKDIR service
COPY target/funnel-mouth .
RUN sudo chown -R $USER .

EXPOSE ${LISTEN_PORT}

ENTRYPOINT ./funnel-mouth --port=${LISTEN_PORT} --stemSubUrl=tcp://${STEM_SUB_HOST}:5000 --vverbose
```

Docker Containers: Runtime Container

A small tidy runtime environment enables faster deployments.

```
# Light-weight runtime environment.
FROM alpine:3.15

# Numerous shared libraries needed in the
environment.
RUN apk add ldc-runtime
RUN apk add openssl
RUN apk add musl
RUN apk add zlib
```

```
# Install AWS CLI tools for use with ECR
RUN apk add --no-cache aws-cli
RUN aws --version

# Create a build user with reduced privileges.
ARG USER=default
ENV HOME /home/$USER

RUN apk add --update sudo
RUN adduser -D $USER \
        && echo "$USER ALL=(ALL) NOPASSWD: ALL" >
/etc/sudoers.d/$USER \
        && chmod 0440 /etc/sudoers.d/$USER

USER $USER
WORKDIR $HOME
```
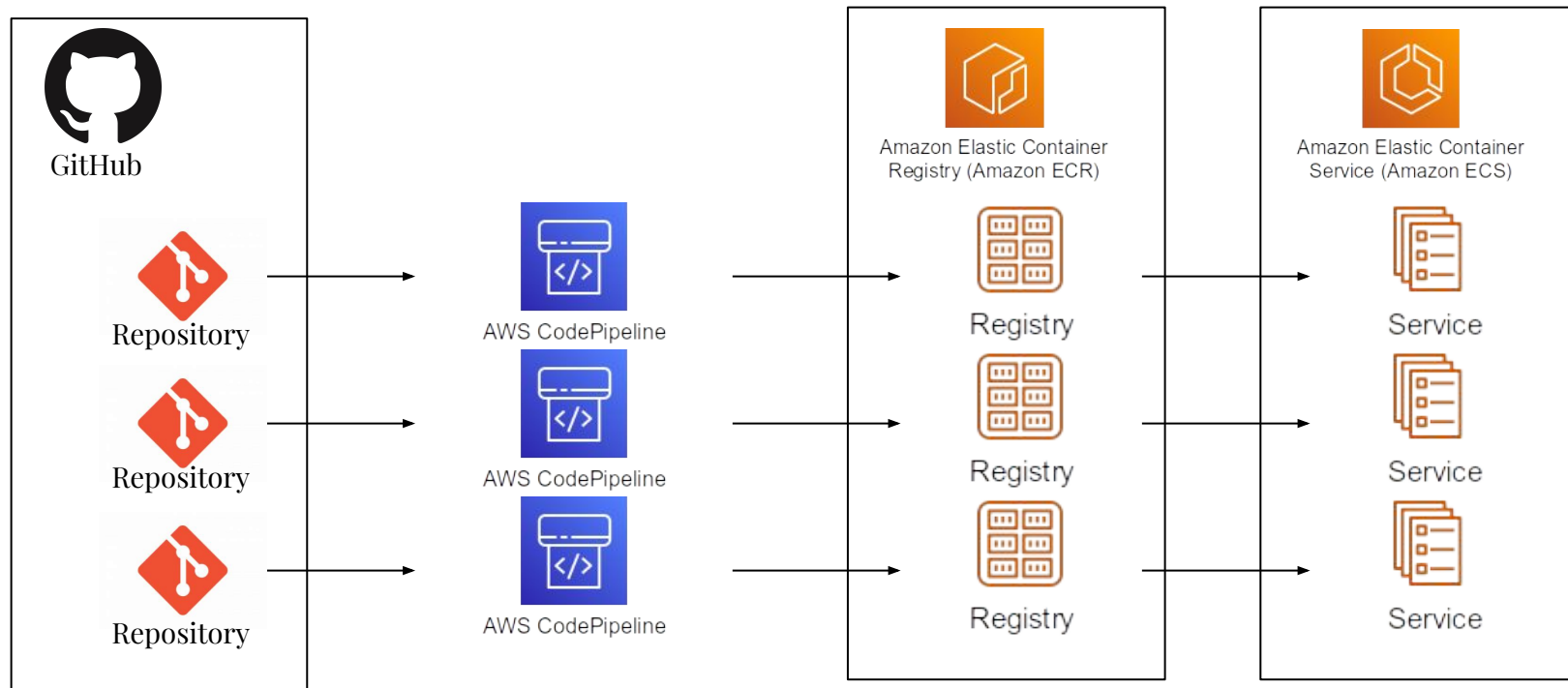
Deployment

Example Deployment Flow

Infrastructure and Deployment

If your goal is to lower costs, keep costs in mind when choosing how to deploy!

For example, should an AWS setup use AWS Elastic Kubernetes Service (EKS) or AWS Elastic Container Service (ECS)?

In terms of price:

- EKS has a base-price of $0.10 per hour, i.e. $72 per month, plus EC2/Fargate.
- ECS has no such base price, only the cost of EC2/Fargate resources.

Right Tools for the Job

If you plan on deploying a project, the tools being used should match the skills of the people doing the work.

[Kubernetes](#) has excellent, albeit lengthy and complicated documentation. Deployment is done via YAML files, which are constructed very carefully.

If you are short on people and do not have dedicated DevOps staff, consider something like [AWS Cloud Development Kit](#) (CDK).

CDK consists of programming APIs that generate config files for Kubernetes, ECS, Terraform, etc.

Program your deployment in a familiar environment with development tools.

Create Reuseable Functions for Standard Configuations

```java
// Constructs a service that follows all the standard rules described in {@link StdServiceSettings}.
public FargateService constructStdService(Cluster cluster, ApplicationListener applicationListener, StdServiceSettings settings) {
    // A FargateService is a load-balanced set of Tasks, each of which are defined by a TaskDefinition.
    FargateTaskDefinition taskDefinition = FargateTaskDefinition.Builder.create(this, settings.name + "-td")
        .cpu(settings.cpu)
        .memoryLimitMiB(settings.memoryLimitMiB)
        .build();

    // Define and add a project container and to the TaskDefinition.
    ContainerDefinition stdContainerDefinition =
        taskDefinition.addContainer(settings.name, ContainerDefinitionOptions.builder()
        .image(ContainerImage.fromEcrRepository(
            Repository.fromRepositoryName(this, settings.name + "-ci", settings.name),
            "latest"))
        // Multiple containers can be in a host, map the host to the containers.
        .portMappings(List.of(
            PortMapping.builder()
                .hostPort(8080)
                .containerPort(8080)
                .build()))
        // ...
        .build());

    FargateService fargateService = FargateService.Builder.create(this, settings.name + "-service")
        .serviceName(settings.name)
        .cluster(cluster)         // Required
        .securityGroups(settings.securityGroup != null ? List.of(settings.securityGroup) : null)
        .desiredCount(settings.desiredCount)  // Default is 1
        .taskDefinition(taskDefinition)
```

Example Deployment Config in Java

```java
FargateService mouthService =
    constructStdService(
        networkStack.getEcsCluster(), loadBalancerStack.getHttpsListener(),
        StdServiceSettings.builder()
            .name("funnel-mouth")
            .securityGroup(networkStack.getSecurityGroup())
            .hostname(hostname)
            .urlPrefixes(List.of("/users/*/data/*"))
            .httpRequestMethods(List.of("POST"))
            .environment(Map.of(
                "JWT_PUBLIC_KEY_URL", "https://auth.funnel-labs.io/auth/realms/funnel",
                "ADMIN_REST_URL", "http://funnel-admin.local:8080"))
            .cpu(512)
            .memoryLimitMiB(1024)
            .desiredCount(2)
            .build());
```

The important point is: **code-completion, documentation, functions, and other programming features can lower the cost of setting up infrastructure**.

Concluding Remarks

- D can deliver significantly higher performance and lower costs than higher-level languages like Java, Python, JavaScript, Ruby, etc.
- D is is very feature rich, making it especially suitable for bringing in libraries from other programming languages.
- D has a good balance of feature to help speed up programming (like garbage collection) without sacrificing much for performance.
- **However**, D suffers from a smaller community and tooling needs some love.

Thank you! Questions? Feedback?