Serverless, Well Actually...

Decoupled Drupal Days 2018

Bob Kepford

About Mediacurrent

Who We Are

Mediacurrent is a **full-service digital agency** that implements world class **open source software development, strategy and design** to achieve defined goals for **enterprise organizations seeking a better return on investment**.

Serverless. What is it?

...cloud-computing execution model in which the cloud provider acts as the server, dynamically managing the allocation of machine resources. Pricing is based on the actual amount of resources consumed by an application, rather than on pre-purchased units of capacity.

– *Wikipedia*

“The phrase “serverless” doesn’t mean servers are no longer involved. It simply means that developers no longer have to think that much about them.”

~ Ken Fromm

So there IS a Server!

Yeah. The Name. Whatever.

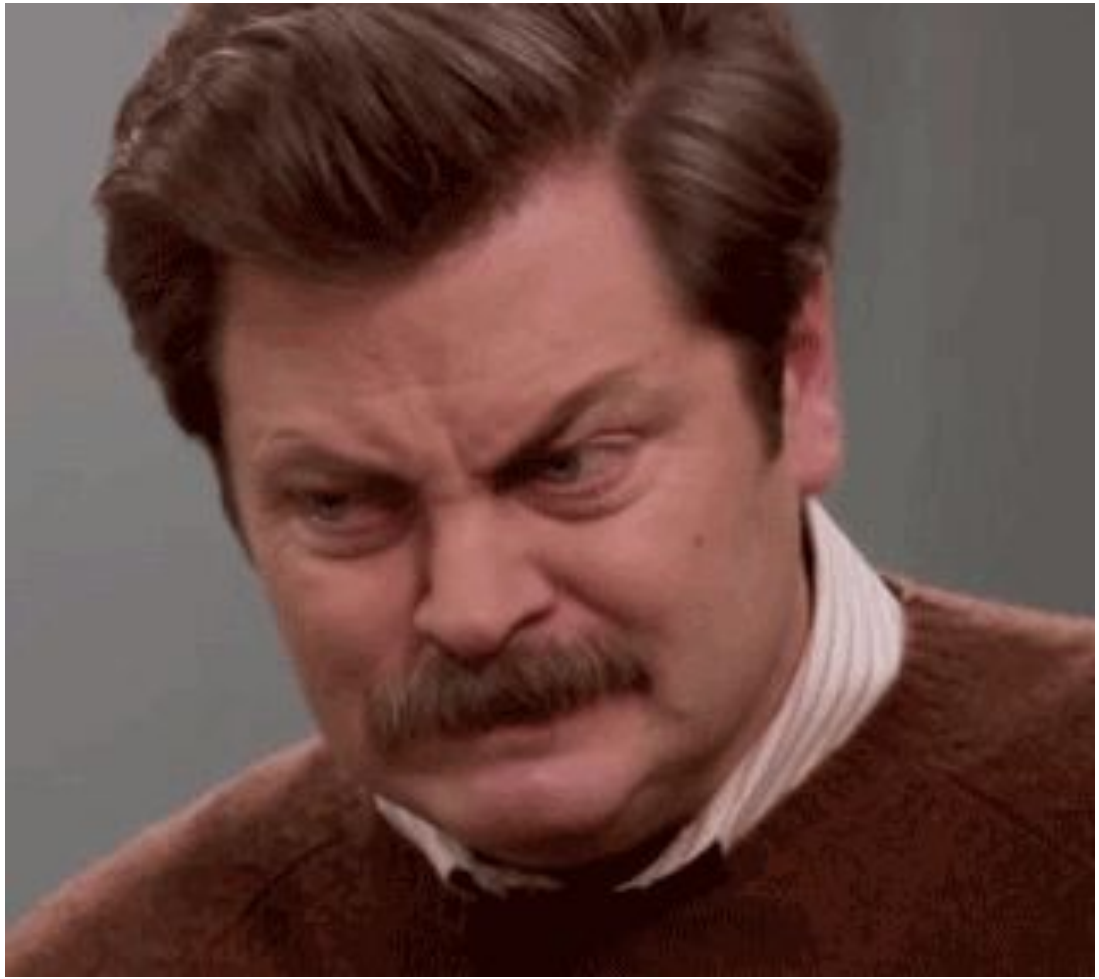
- No servers to manage
- Only pay for what you use
- Many services available, some you already use.

~ Bob Kepford

Mike Roberts' Definition of Serverless

1. No management of server hosts or server processes
2. Self auto-scale and auto-provision based on load
3. Costs based on precise usage
4. Performance capabilities defined in terms other than host size/count
5. Implicit high availability

https://www.martinfowler.com/articles/serverless.html#WhatIsServerless



So What's the Big Deal

- Low start up cost
- Predictable pricing
- Web scale
- Write your front-end and back-end in JavaScript
- Built for microservices approaches

Give Me specifics

- HTTP Endpoints (Functions/Business Logic)
- Data stores (SQL/NoSQL)
- Files
- Email
- Queuing
- Authentication
- I'm running out of screen space

How'd We Get Here?

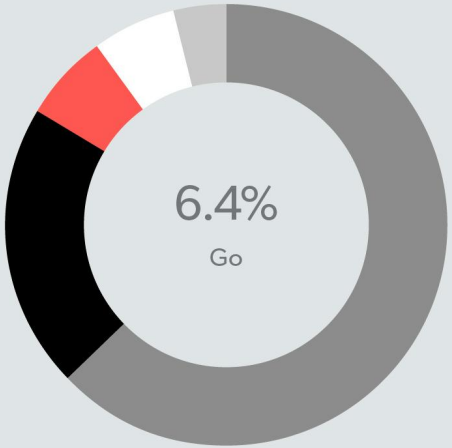
1. Microservices
2. AWS
3. PaaS
4. Kubernetes
5. Serverless

Service Providers

- Amazon (AWS)
- Google (Cloud & Firebase)
- Microsoft (Azure)
- IBM (Uses Apache OpenWhisk which is OSS)
- Some that build on top of AWS like Webtask

Serverless Languages

Languages used for serverless development



- **62.9%** Node.js
- **20.8%** Python
- **6.4%** Go
- **6.1%** Java
- **3.8%** C#

serverless

serverless.com

Common Uses

- Great for static sites - contact forms
- Great for image processing
- Ops / Infrastructure
- IoT
- Data processing
- Serverless at the edge
- Avoiding high upfront costs for servers

When does it makes sense to use it

- Discrete API functionality
- Scale is a concern
- Cost is a concern
- Maintenance is a concern

When does it not make sense

- Tight timeline & inexperienced team
- Execution time is critical
- Need Co-located resources
- Pricing: Some high traffic situations. Mileage varies

How much does it cost?

- It depends...
- Calculators
- It can be cost effective
- Set billing limits and notifications

Vendor Lock-In

- Less of an issue due to the microservices nature
- Integration with the platform's Database services and unique features is an issue.

Frameworks and Tools

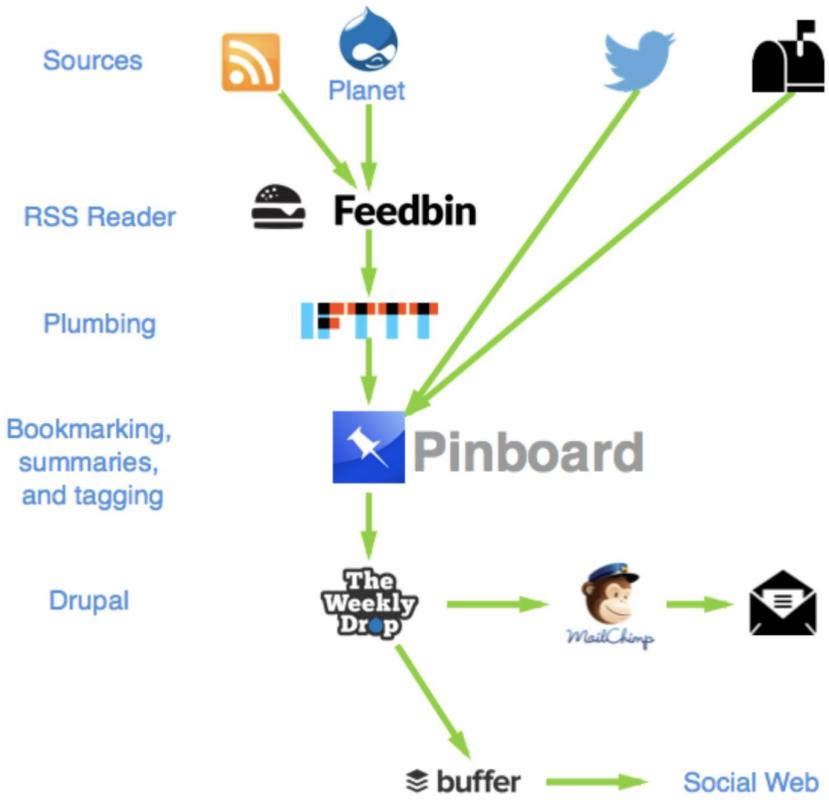
- Serverless framework - serverless.com
- SAM (AWS Serverless Application Model) - github.com/awslabs/serverless-application-model

Serverless Framework

- Standardizes project structure
- Deployment
- Rich plugin ecosystem
- Tries to make migration from one provider to another easier

Real World Use Case

TheWeeklyDrop


Sources
Planet
RSS Reader
Feedbin
Plumbing
IFTTT
Bookmarking, summaries, and tagging
Pinboard
Drupal
The Weekly Drop
MailChimp
buffer
Social Web

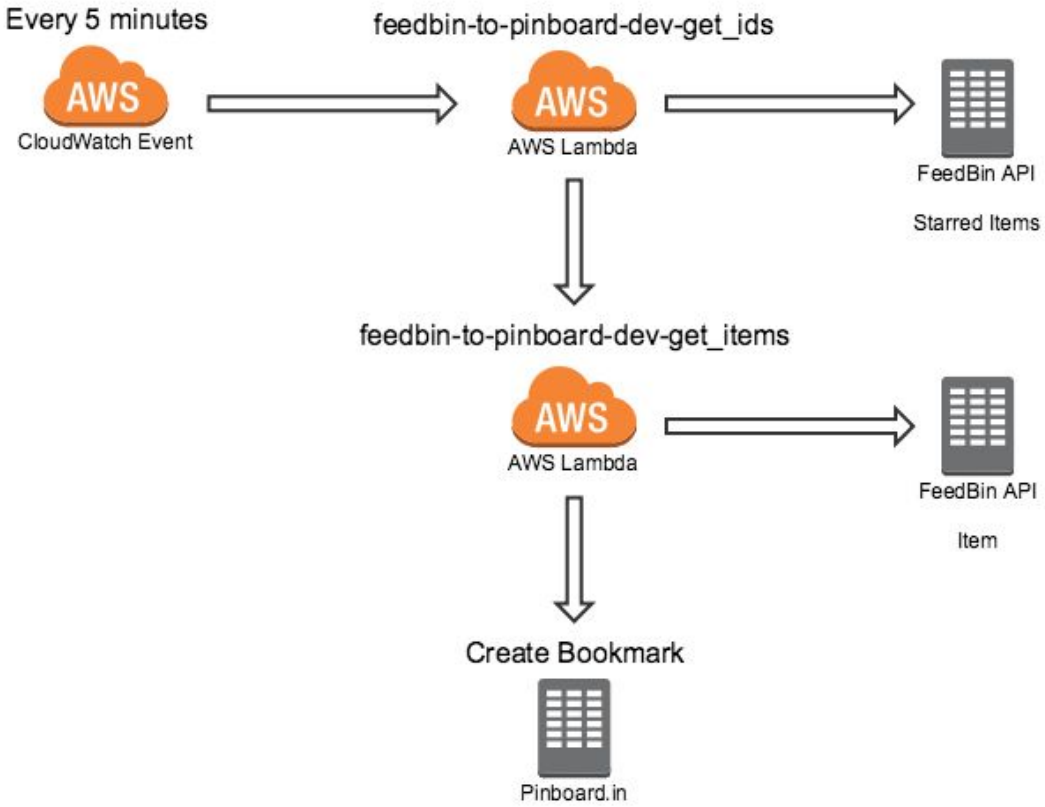

SERVERLESS


aws


node
JS

Feedbin to Pinboard

github.com/kepford/serverless-feedbin-to-pinboard


Every 5 minutes
AWS
CloudWatch Event
feedbin-to-pinboard-dev-get_ids
AWS
AWS Lambda
FeedBin API
Starred Items
feedbin-to-pinboard-dev-get_items
AWS
AWS Lambda
FeedBin API
Item
Create Bookmark
Pinboard.in

Project Directory

```
feedbin-to-pinboard/
| node_modules/
| utils/
| | createBookmark.js
| | feedbinRequest.js
| | getFeedbinEntries.js
| | getFeedbinEntryIds.js
| | unstarFeedbin.js
| LICENSE
| README.md
| example.env.js
| handler.js
| package.json
| serverless.yml
```

serverless.yml

```yaml
service: feedbin-to-pinboard

plugins:
  - serverless-offline
  - serverless-offline-scheduler

provider:
  name: aws
  runtime: nodejs6.10
  stage: ${opt:stage, 'dev'}
  region: us-west-1
  profile: default
  environment:
    REGION: ${self:provider.region}
    STAGE: ${self:provider.stage}
  iamRoleStatements:
    - Effect: Allow
      Action:
        - "lambda:InvokeFunction"
      Resource: "*"
package:
  include:
    - utils/**
functions:
  get_ids:
    handler: handler.getIds
    events:
      - schedule:
          name: check-feedbin
          description: 'Checks the feedbin url every 5 minutes'
          rate: rate(5 minutes)
  get_items:
    handler: handler.getItems
    events:
      - http:
          path: item
          method: post
```

serverless.yml

```
service: feedbin-to-pinboard

plugins:
  - serverless-offline
  - serverless-offline-scheduler

provider:
  name: aws
  runtime: nodejs6.10
  stage: ${opt:stage, 'dev'}
  region: us-west-1
  profile: default
  environment:
    REGION: ${self:provider.region}
    STAGE: ${self:provider.stage}
  iamRoleStatements:
    - Effect: Allow
      Action:
        - "lambda:InvokeFunction"
      Resource: "*"
```

serverless.yml

```yaml
package:
  include:
    - utils/**
functions:
  get_ids:
    handler: handler.getIds
    events:
      - schedule:
          name: check-feedbin
          description: 'Checks the feedbin url every 5 minutes'
          rate: rate(5 minutes)
  get_items:
    handler: handler.getItems
    events:
      - http:
          path: item
          method: post
```

package.json

```
{
  "name": "feedbin-to-pinboard",
  "version": "1.0.0",
  "main": "handler.js",
  "scripts": {
    "sls": "serverless",
    "offline": "sls offline start",
    "test": "echo \"Error: no test specified\" && exit 1"
  },
  "keywords": [],
  "author": "",
  "license": "ISC",
  "dependencies": {
    "aws-sdk": "^2.275.1",
    "node-pinboard": "^1.0.0",
    "request": "^2.87.0",
    "request-promise": "^4.2.2"
  },
  "devDependencies": {
    "eslint": "^4.19.1",
    "serverless": "^1.27.3",
    "serverless-offline": "^3.25.4",
    "serverless-offline-scheduler": "^0.3.4"
  },
  "description": "Takes a feed of Feedbin entries and creates Pinboard bookmarks."
}
```

handler.js getIds()

```
1  'use strict';
2
3  const getFeedbinEntryIds = require('./utils/getFeedbinEntryIds.js');
4  const getFeedbinEntries = require('./utils/getFeedbinEntries.js');
5  const createBookmark = require('./utils/createBookmark.js');
6  const unstarFeedbin = require('./utils/unstarFeedbin.js');
7  const config = require('./.env.js');
8
9  module.exports.getIds = (event, context, callback) => {
10
11   // Get entries from Feedbin.
12   // Calls getItem lambda function.
13   getFeedbinEntryIds(event);
14 };
```

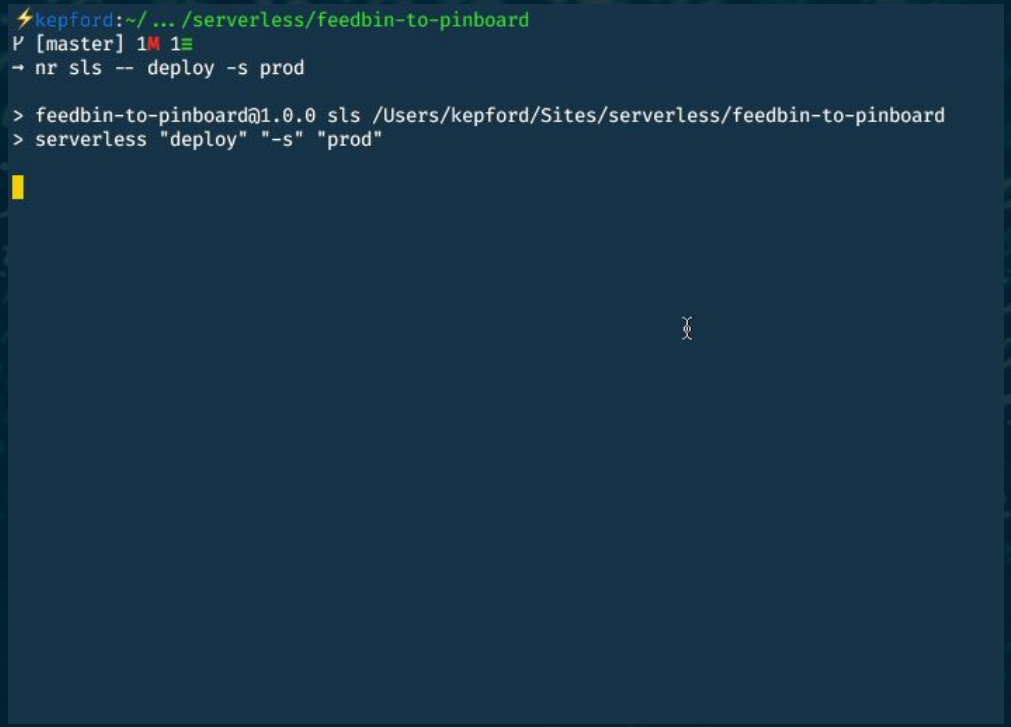
handler.js
getItems()

```
module.exports.getItems = (event, context, callback) => {
  const ids = JSON.parse(event);
  const entryContent = getFeedbinEntries(ids);
  entryContent.then(data => {
    const items = data.map(item => JSON.parse(item));
    items.forEach(function(item) {
      const options = {
        description: item.title,
        url: item.url,
        toread: config.pinboard.hasOwnProperty('toread') ? config.pinboard.toread : 'no',
        tags: config.pinboard.hasOwnProperty('tags') ? config.pinboard.tags : '',
        shared: config.pinboard.hasOwnProperty('shared') ? config.pinboard.shared : 'no'
      };
      const bookmark = createBookmark(options, item.id);
      bookmark.then(pbRes=> {

        // Unstar if the bookmark was created.
        if (pbRes.result_code === 'done') {
          const unstar = unstarFeedbin({
            starred_entries: [pbRes.id]
          });
          unstar.then(fbRes => {
            callback(null, {
              statusCode: 200,
              body: JSON.stringify({
                message: `Created bookmark and unstared Item: ${fbRes.id}`
              })
            });
          });

        }
        else {
          callback(null, {
            statusCode: 200,
            body: JSON.stringify({
              message: `Response code ${pbRes.result_code} for Feedbin ID: ${pbRes.id}`
            })
          });
        }
      });
    });
  });
};
```

Deployment

```
sls deploy
sls deploy -s prod
sls deploy -f functionName -s dev
```

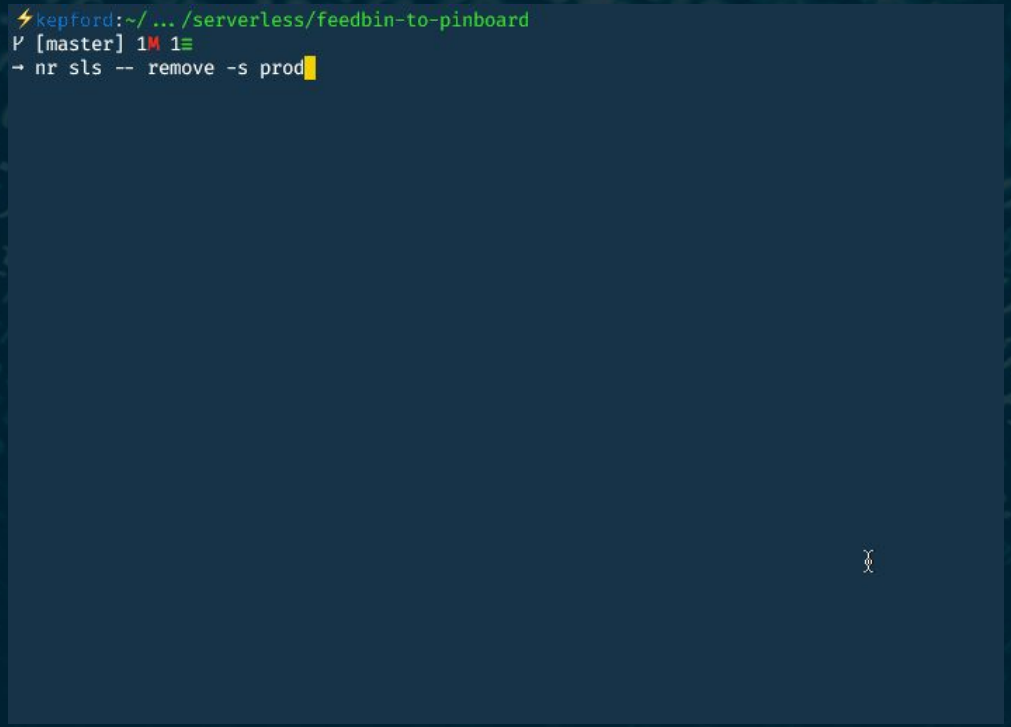
Serverless deploy



Removal

```
sls remove
sls remove -s prod
```

Removal


⚡kepford:~/ .../serverless/feedbin-to-pinboard
ᚠ [master] 1M 1≡
→ nr sls -- remove -s prod

Logging? Yep!

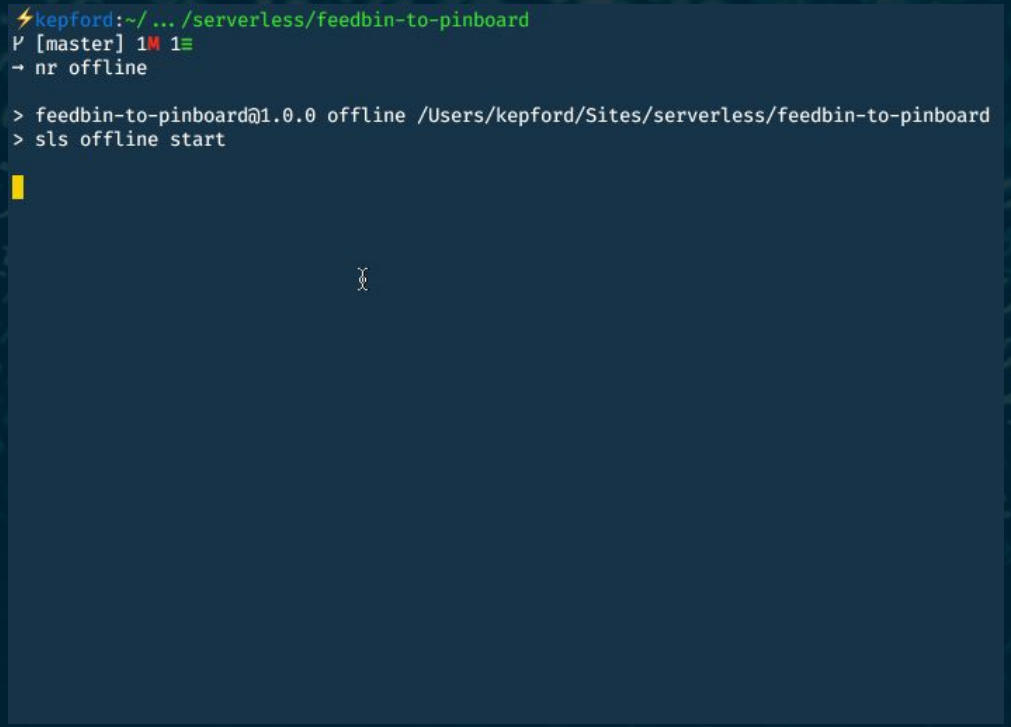
```
sls logs -f hello -s dev -t
```

Tail the logs on dev for function named hello.

Local Development? Yep

```
sls offline start
sls invoke -f functionName -l
```

Serverless Offline



Resources

- Think FAAS with TREK10 - Podcast
- The Serverless Framework with Node.js & AWS - Udemy Course
- Serverless Status - Newsletter
- Serverless Chronicle - Newsletter
- The Power of Serverless - Site for Frontend Developers


mediacurrent
Thank you!
@Mediacurrent
facebook.com/mediacurrent
Mediacurrent.com

Today's Team

Josh Linard
VP, Sales
Executive Sponsor

Dave Terry
Partner, Client Services

Bill Shaouy
Senior Business Analyst

Matt Davis
Lead Architect
Emerging Technology