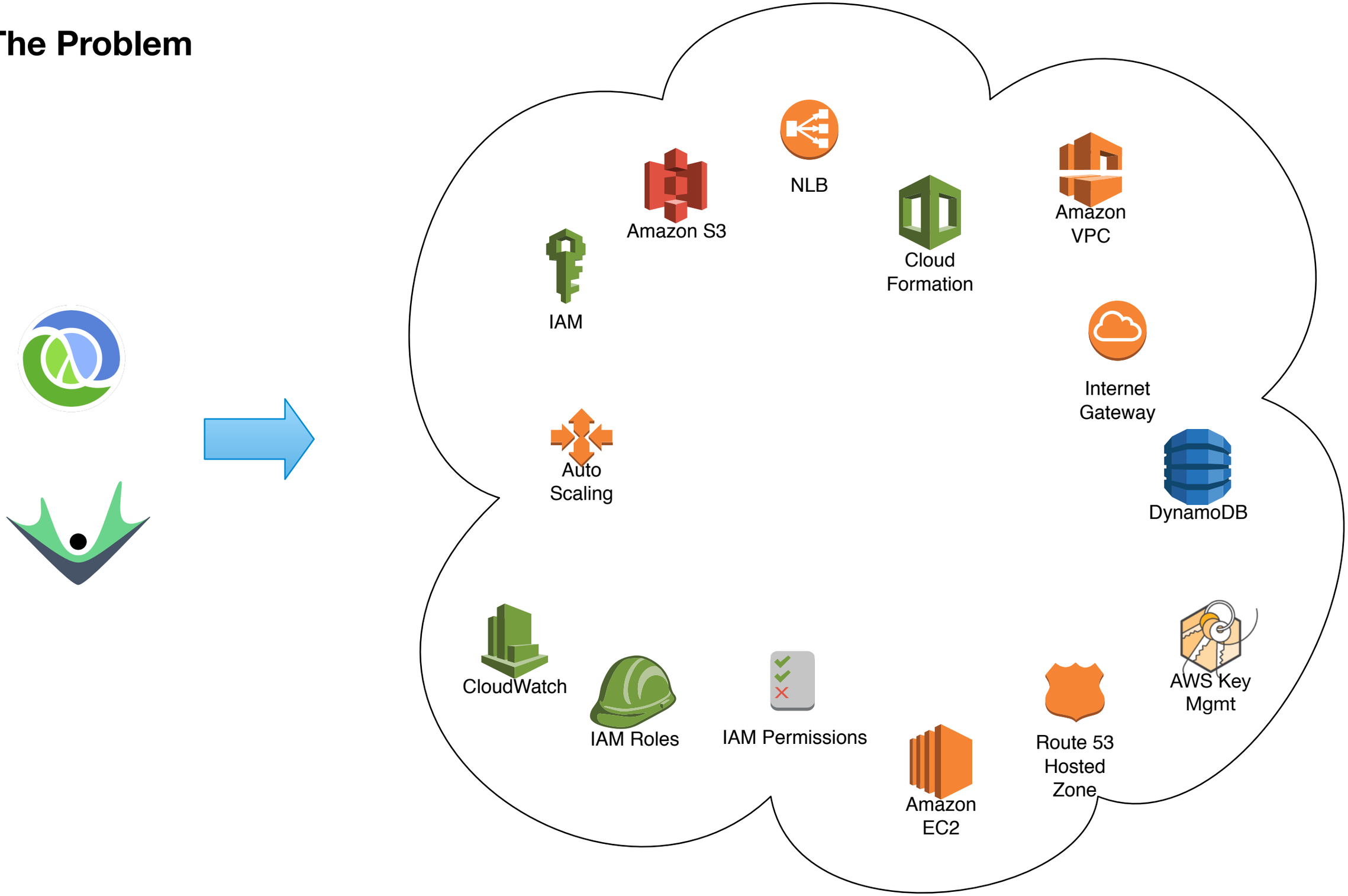
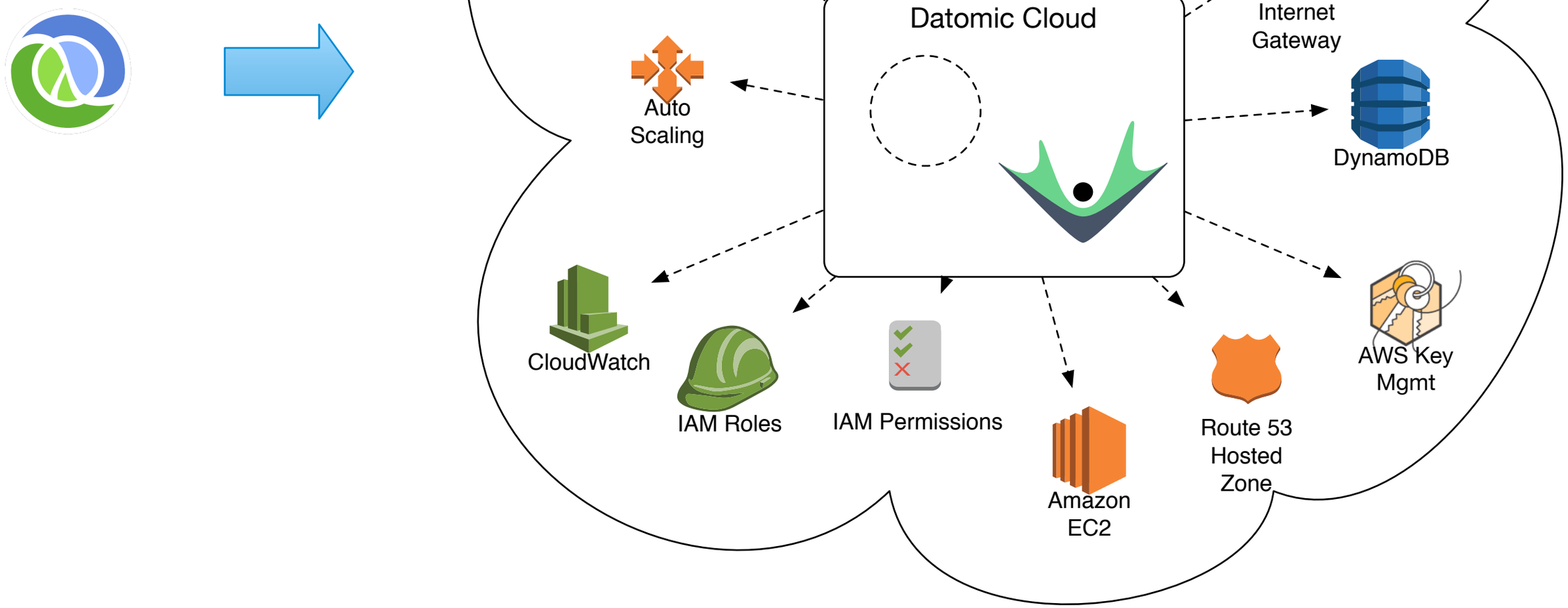
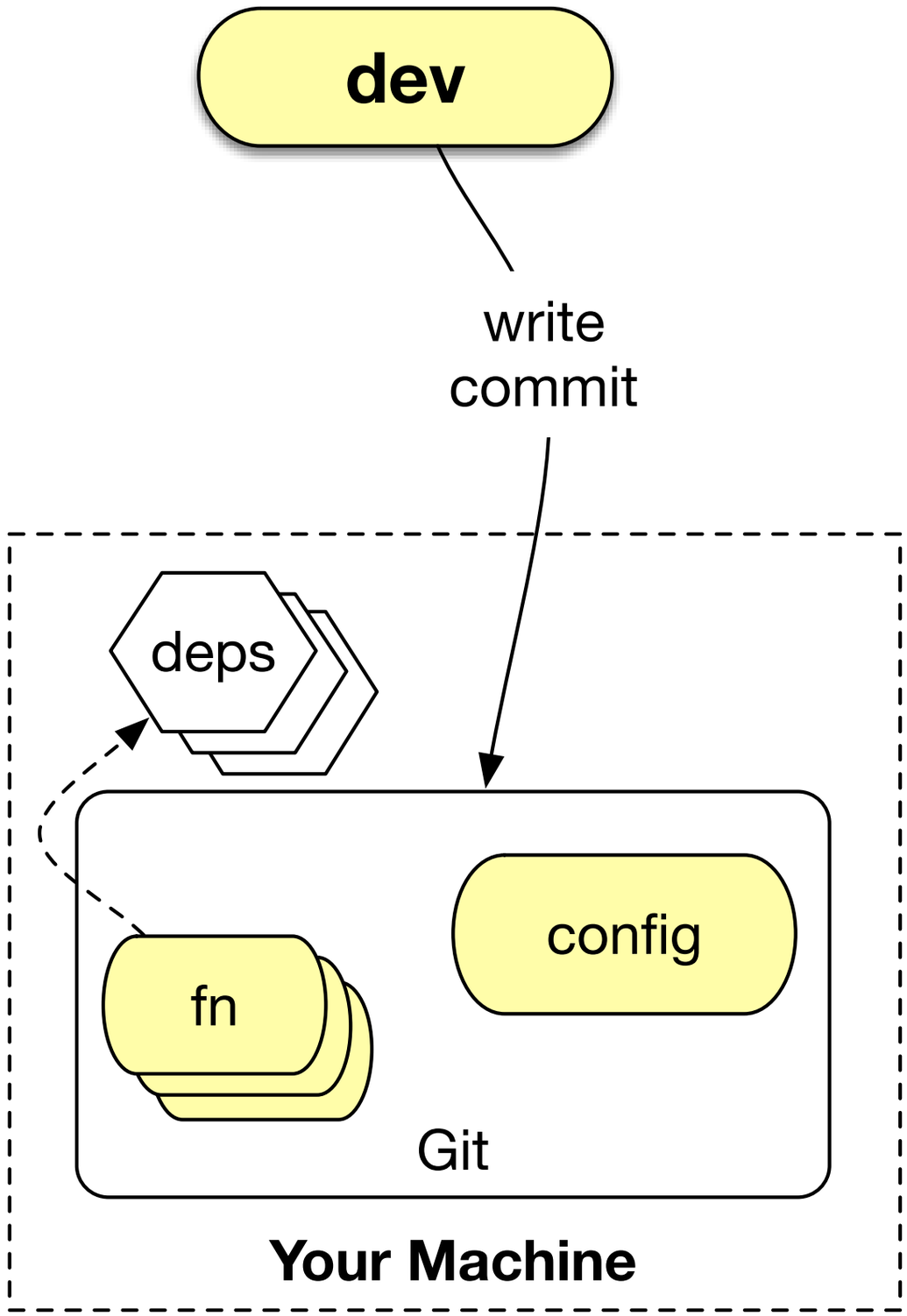
Datomic Ions

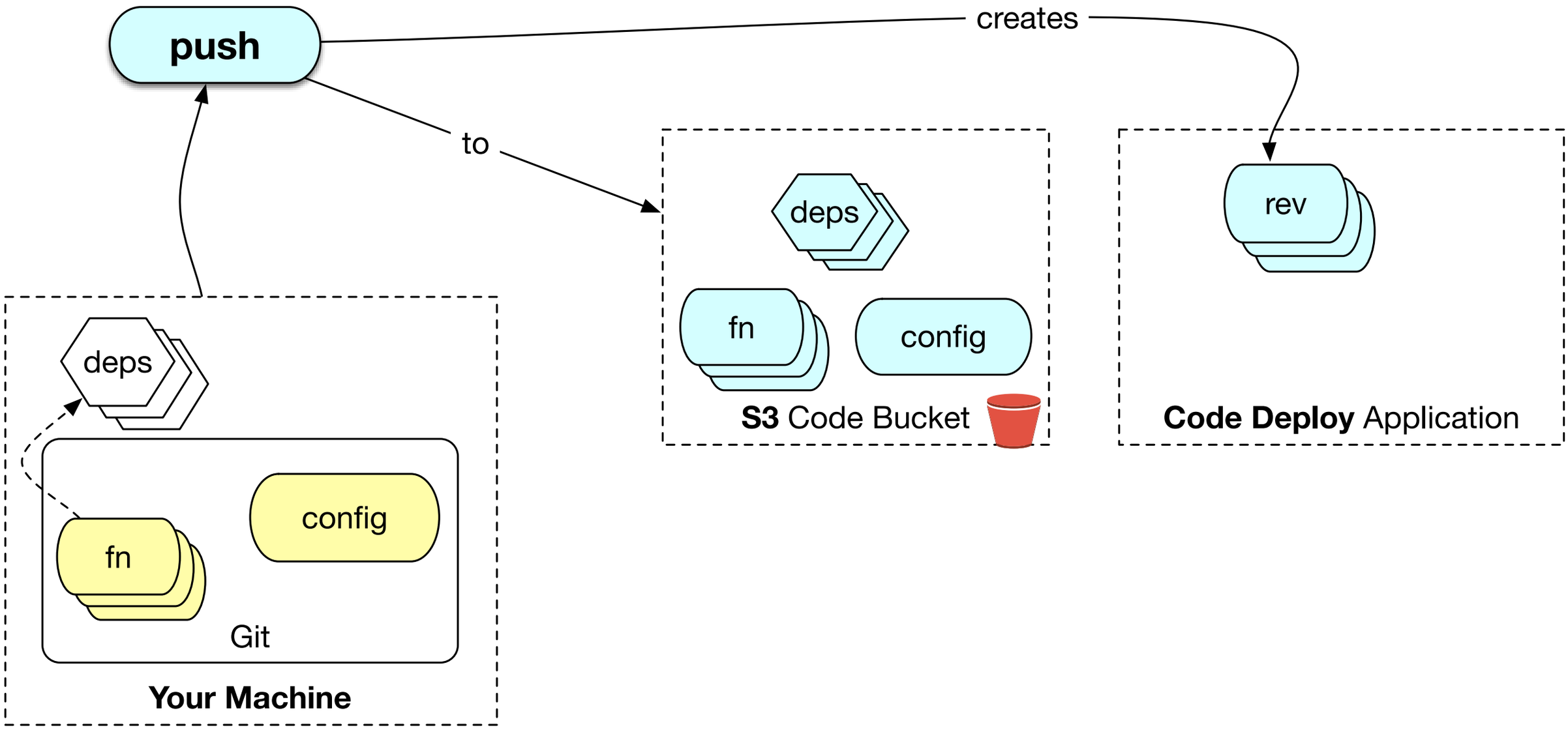
@stuarthalloway

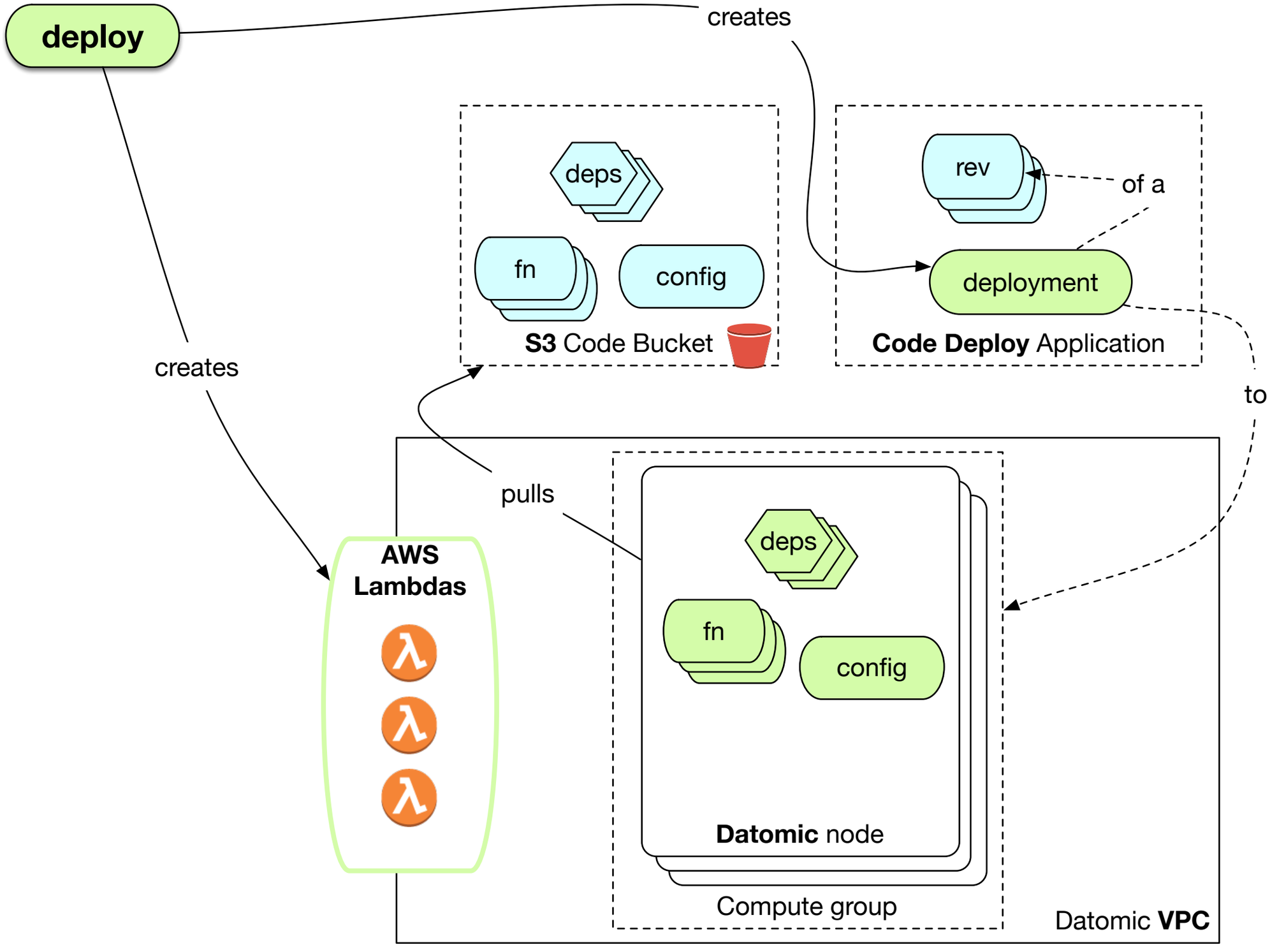
The Problem

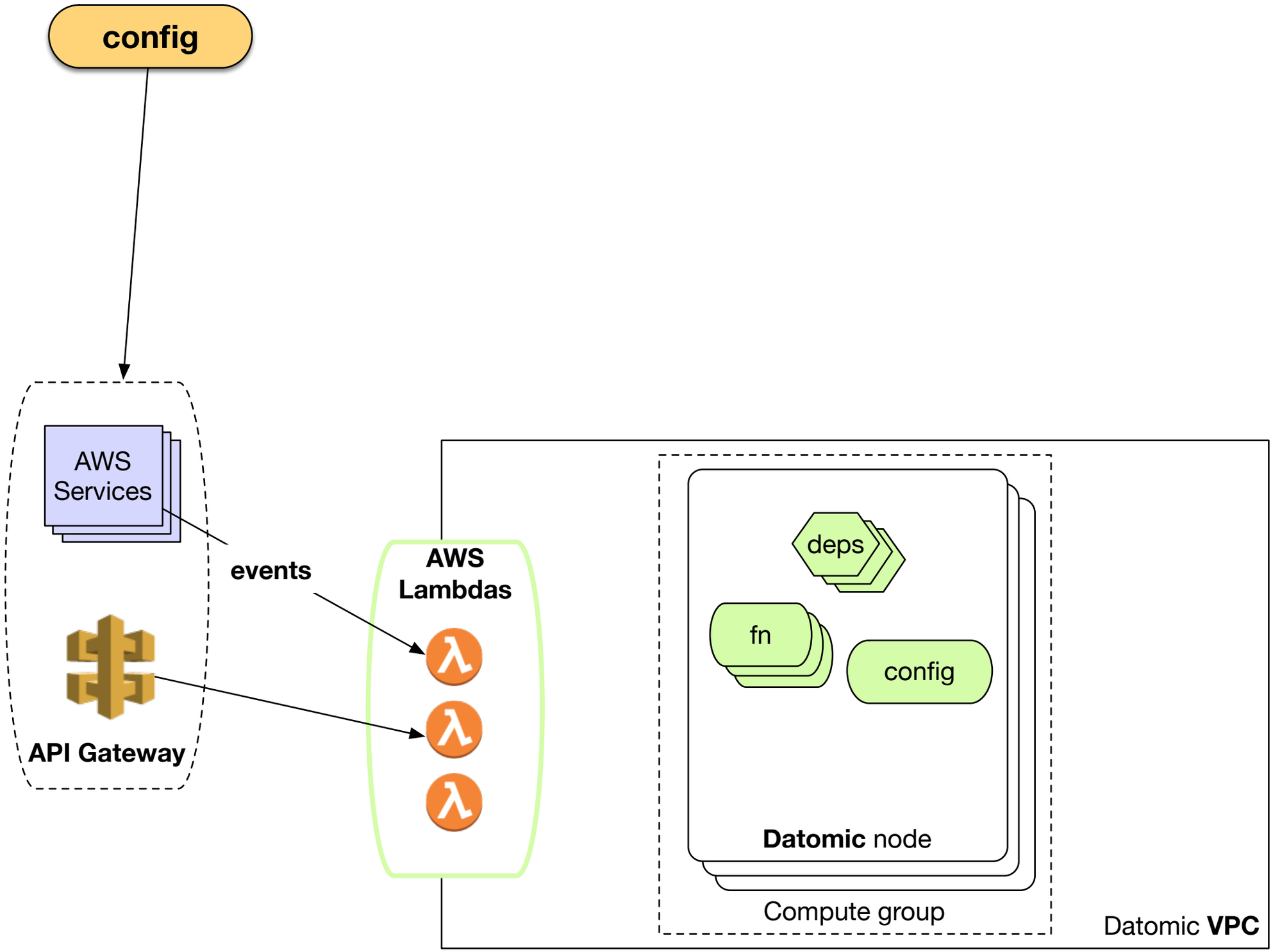

NLB
Amazon S3
Amazon VPC
IAM
Cloud Formation
Internet Gateway
Auto Scaling
DynamoDB
CloudWatch
IAM Roles
IAM Permissions
AWS Key Mgmt
Route 53 Hosted Zone
Amazon EC2

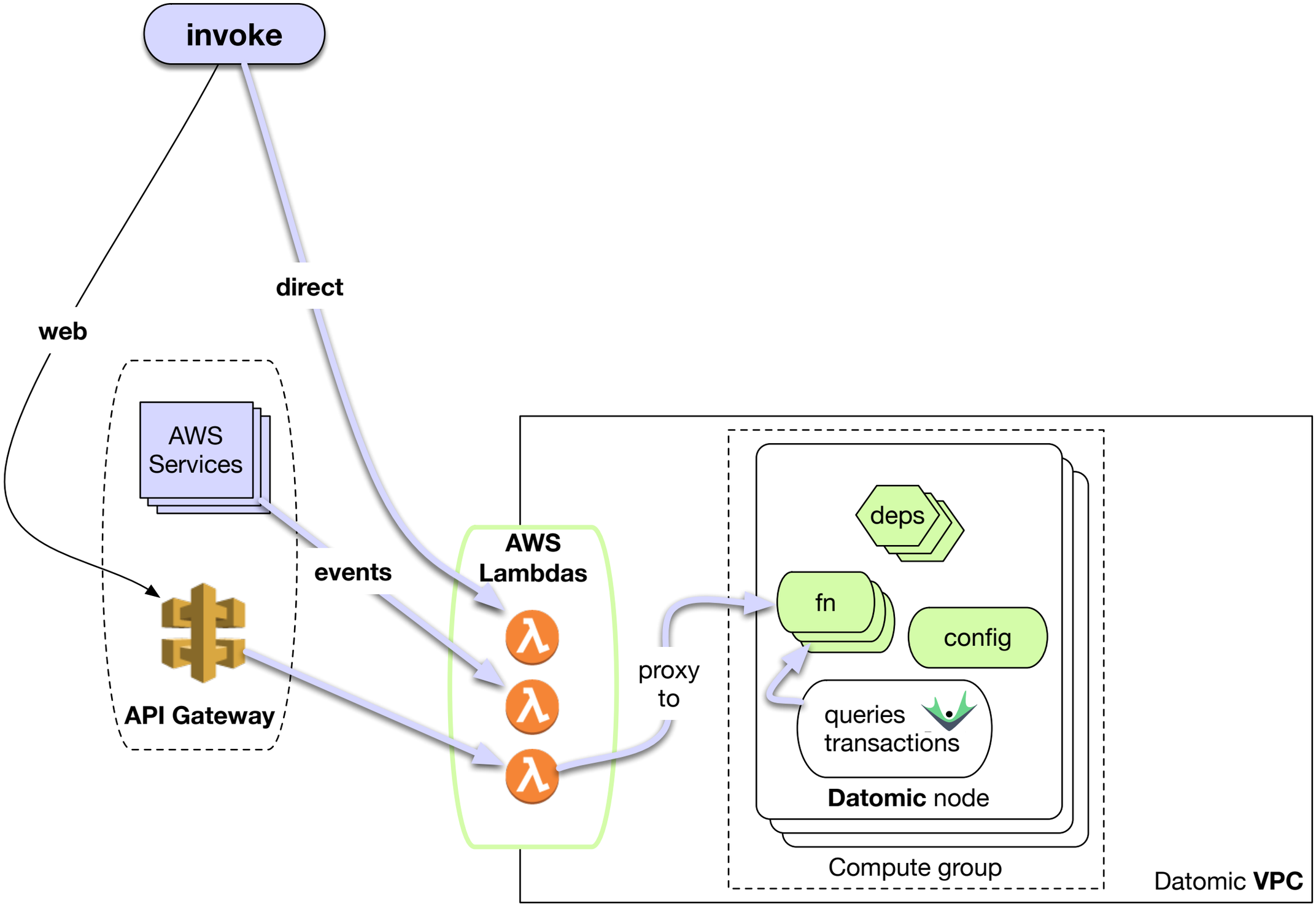

NLB
Amazon S3
Amazon VPC
IAM
Cloud Formation
Datomic Cloud
Internet Gateway
Auto Scaling
DynamoDB
CloudWatch
IAM Roles
IAM Permissions
AWS Key Mgmt
Amazon EC2
Route 53 Hosted Zone


dev
write
commit
deps
config
fn
Git
Your Machine

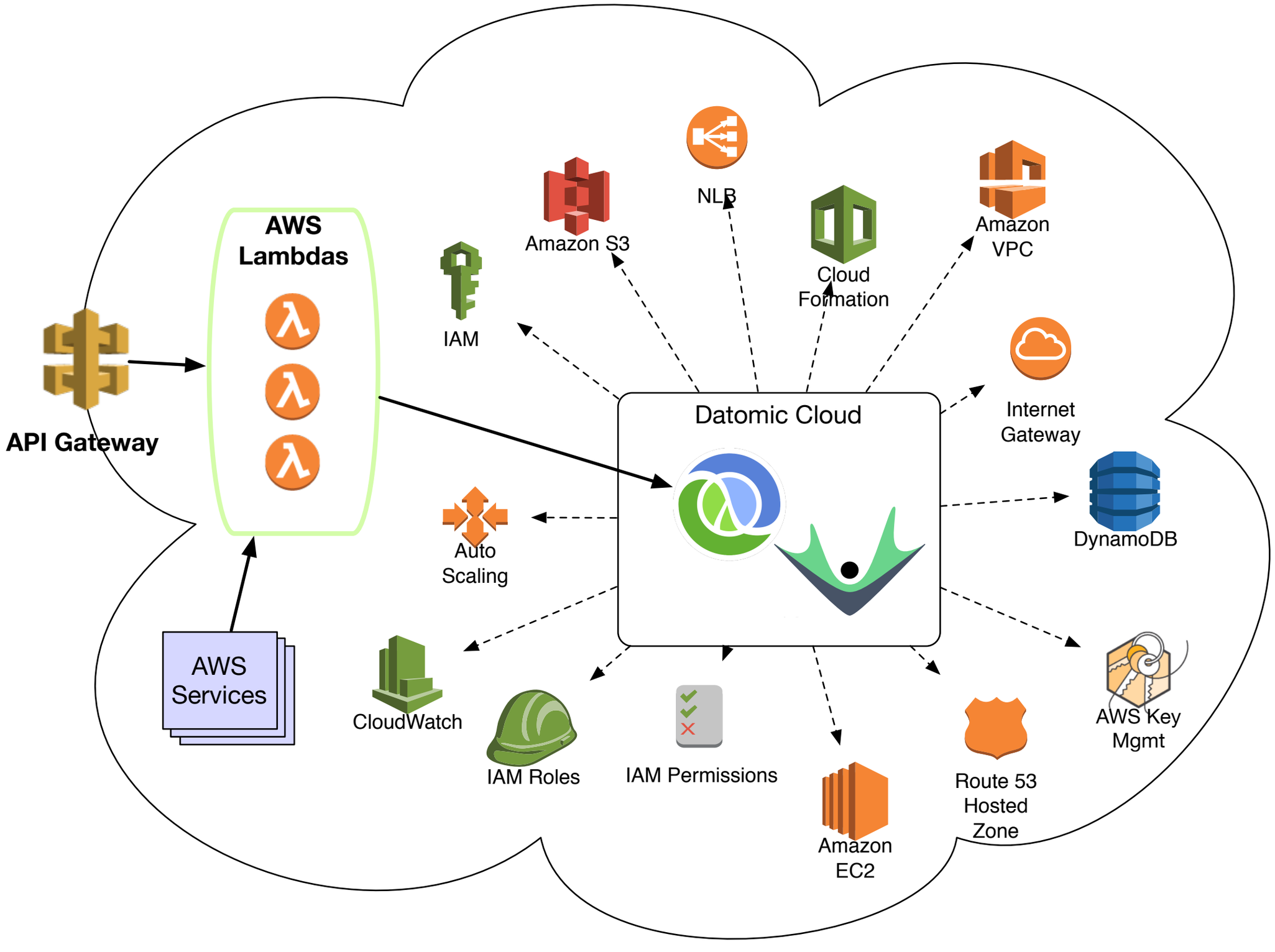

push
creates
to
deps
rev
fn
config
S3 Code Bucket
Code Deploy Application
deps
config
fn
Git
Your Machine


deploy
creates
creates
deps
fn
config
S3 Code Bucket
rev
of a
deployment
Code Deploy Application
to
pulls
AWS
Lambdas
deps
fn
config
Datomic node
Compute group
Datomic VPC


config
AWS
Services
events
AWS
Lambdas
API Gateway
deps
fn
config
Datomic node
Compute group
Datomic VPC


invoke
web
direct
AWS
Services
events
API Gateway
AWS
Lambdas
proxy
to
deps
fn
config
queries
transactions
Datomic node
Compute group
Datomic VPC

Datomic Ions



https://github.com/Datomic/ion-event-example

slackbot that also responds to AWS CodeDeploy events

DeployStatus APP 12:14 PM

```
|                         :time |        :group |  :app |   :state |          :id |
|-------------------------------+---------------+-------+----------+--------------|
| Thu Jun 14 16:08:10 UTC 2018 | stu-8-compute | stu-8 |   :START | d-FQFA6JPPT |
| Thu Jun 14 16:08:14 UTC 2018 | stu-8-compute | stu-8 |   :START | d-FQFA6JPPT |
| Thu Jun 14 16:08:37 UTC 2018 | stu-8-compute | stu-8 | :SUCCESS | d-FQFA6JPPT |
| Thu Jun 14 16:08:38 UTC 2018 | stu-8-compute | stu-8 |   :START | d-FQFA6JPPT |
| Thu Jun 14 16:09:01 UTC 2018 | stu-8-compute | stu-8 | :SUCCESS | d-FQFA6JPPT |
| Thu Jun 14 16:09:03 UTC 2018 | stu-8-compute | stu-8 | :SUCCESS | d-FQFA6JPPT |
```

stu 12:24 PM

hey @deploystatus how are you? (edited)

dev write ordinary fns

lons take care of

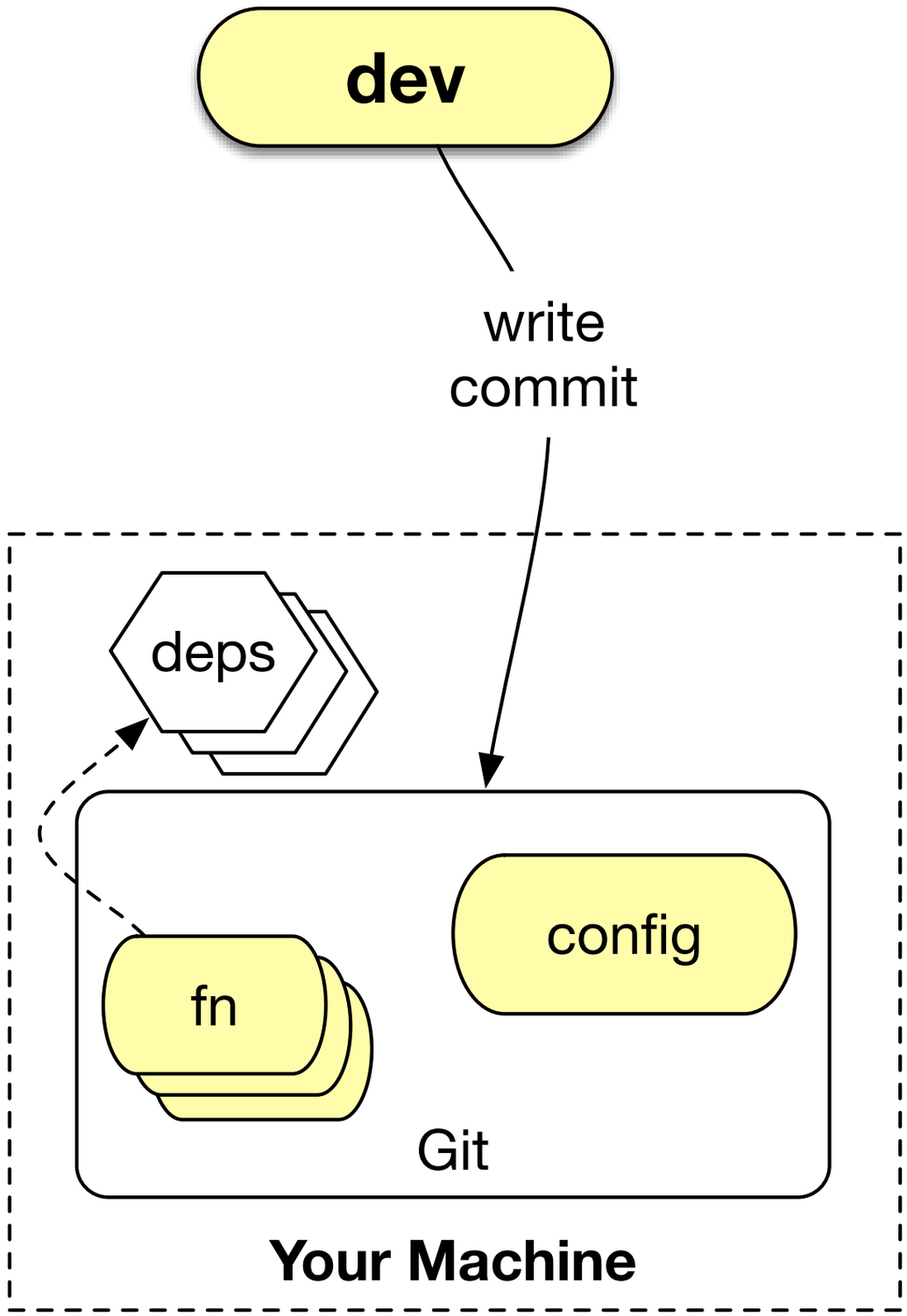
push

deploy

configure

invoke

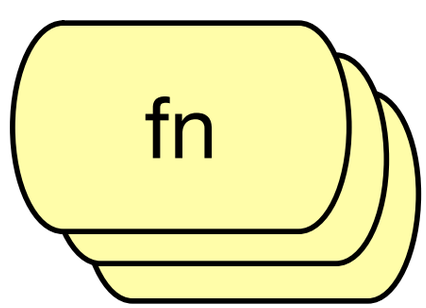
power / reach / agility


dev
write
commit
deps
config
fn
Git
Your Machine

Lambda Ion

```
1 (defn event-handler
2   "Ion that responds to AWS CodeDeploy events. Transacts event
3 info into the database, and posts a notification in Slack."
4   [{:keys [input]}]
5   (let [conn (get-conn)
6         data (->> input
7                   (event->tx code-deploy-event-rules)
8                   (add-refs code-deploy-event-refs))
9         slack-channel (get-config (d/db conn) :slack/channel)]
10    (d/transact conn {:tx-data data})
11    (post-slack-message slack-channel (-> data pr-str code-block))
12    "handled"))
```

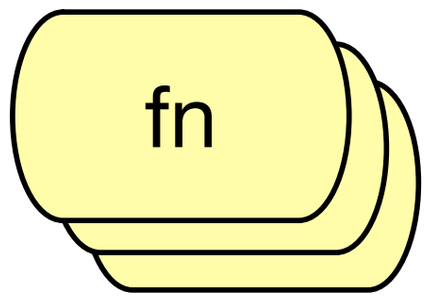
https://github.com/Datomic/ion-event-example/blob/master/src/datomic/ion/event_example.clj



Web Ion

```clojure
(defn slack-event-handler*
  "Web service ion that responds to slack notifications by printing
an org-mode table of recent deploys."
  [{:keys [headers body]}]
  (let [json (-> body io/reader (json/read :key-fn keyword))
        db (d/db (get-conn))
        slack-channel (get-config db :slack/channel)
        verified? (= (get json :token)
                     (get-config db :slack/verification-token))]
    (if verified?
      (if-let [challenge (get json :challenge)]
        {:status 200 :headers {} :body challenge}
        (let [posted (post-slack-message slack-channel (deploys-table))]
          {:status 200 :headers {}}))
      {:status 503 :headers {}})))
```

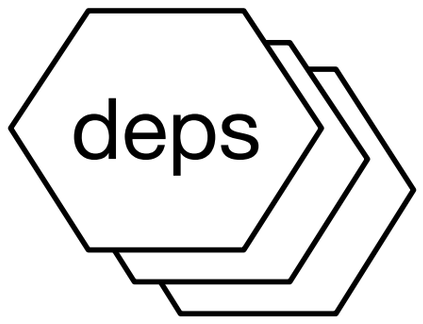
https://github.com/Datomic/ion-event-example/blob/master/src/datomic/ion/event_example.clj



Deps

```
{:paths ["src" "resources"]
 :deps {com.datomic/ion {:mvn/version "0.9.7"}
        org.clojure/data.json {:mvn/version "0.2.6"}
        org.clojure/clojure {:mvn/version "1.9.0"}}
 :mvn/repos {"datomic-cloud"
             {:url "s3://datomic-releases-1fc2183a/maven/releases"}}
 :aliases
 {:dev
  {:extra-deps
   {com.datomic/client-cloud {:mvn/version "0.8.54"}
    com.datomic/ion-dev {:mvn/version "0.9.162"}}}}}
```

https://github.com/Datomic/ion-event-example/blob/master/deps.edn



Config

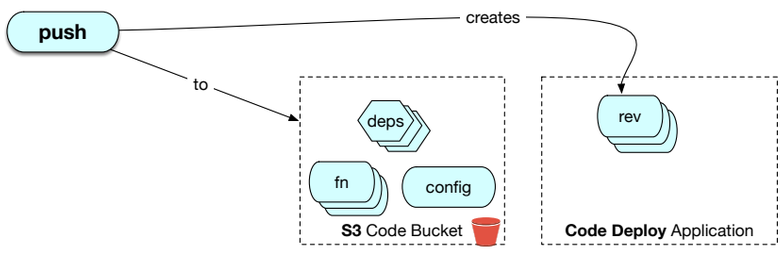
```
{:allow [datomic.ion.event-example/event-handler
         datomic.ion.event-example/slack-event-handler
         ]
 :lambdas {:event-handler
           {:fn datomic.ion.event-example/event-handler
            :description "Handles AWS AutoScaling events"}
           :slack-event-handler
           {:fn datomic.ion.event-example/slack-event-handler
            :integration :api-gateway/proxy
            :description "handles Slack events"}}
 :app-name "stu-8"}
```

https://github.com/Datomic/ion-event-example/blob/master/resources/datomic/ion-config.edn

config

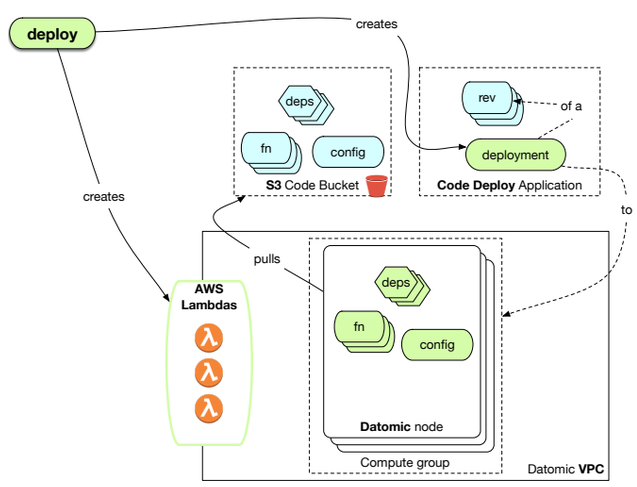
Push and Deploy

```
1 clojure -Adev -m datomic.ion.dev '{:op :push
                                    :uname “stu”}'
```



push

```
2 clojure -Adev -m datomic.ion.dev '{:op :deploy
                                    :group "stu-8-compute"
                                    :uname "stu"}'
```



deploy

CloudWatch Event Rule

Step 1: Create rule

Create rules to invoke Targets based on Events happening in your AWS environment.

Event Source

Build or customize an Event Pattern or set a Schedule to invoke Targets.

(•) Event Pattern ( ) Schedule

Build event pattern to match events by service

Service Name: CodeDeploy

Event Type: State Change

(•) Any detail type ( ) Specific detail type(s)

Select a specific detail type...

Targets

Select Target to invoke when an event matches your Event Pattern or when schedule is triggered.

Lambda function

Function*: stu-8-compute-event-handler

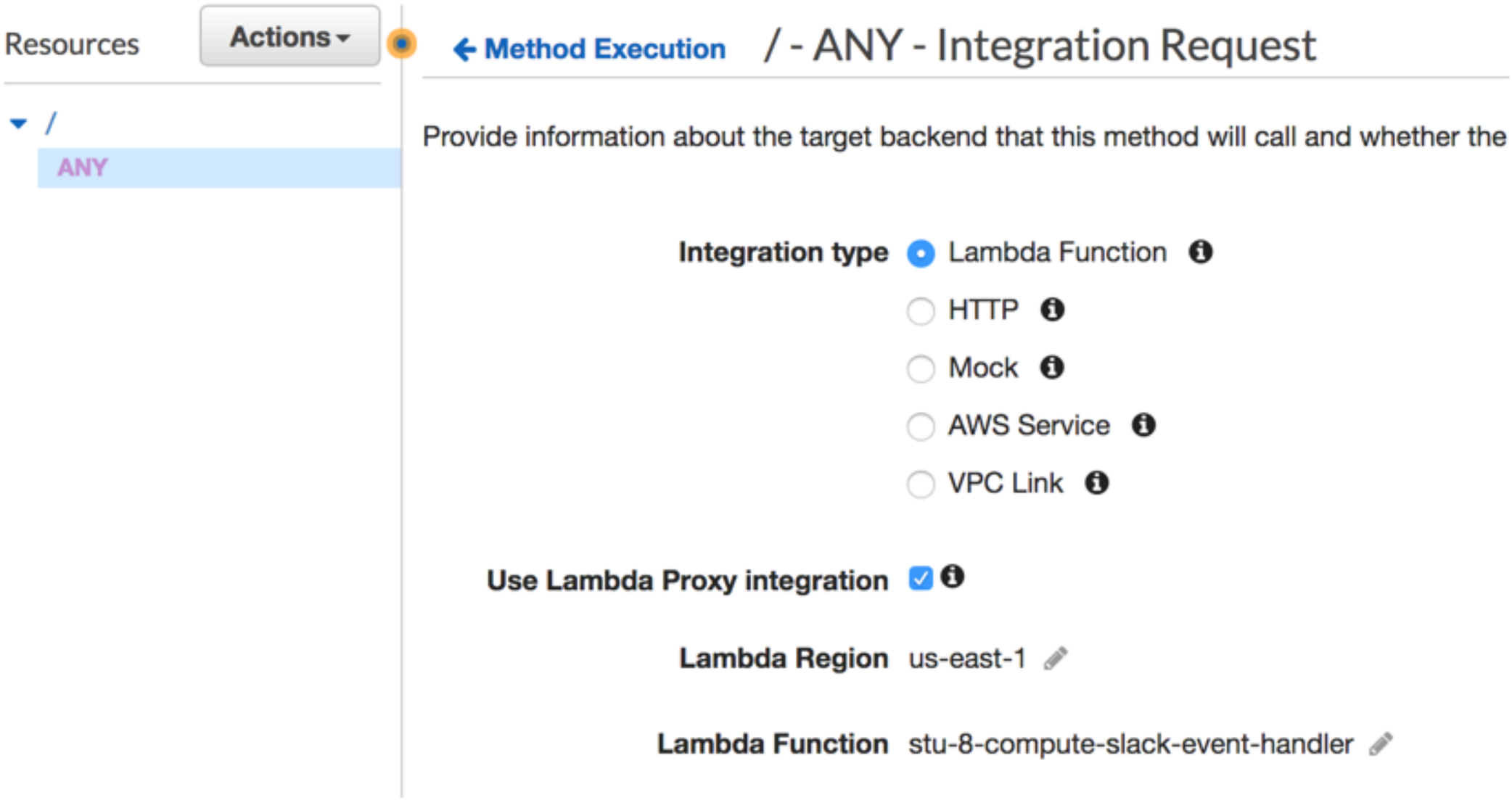
▸ Configure version/alias

▸ Configure input

Add target*

configure

API Gateway Integration



configure

Lambda Invocation

stu-8-compute... Throttle Qualifiers Actions test Test Save

Execution result: succeeded (logs)

Details

The area below shows the result returned by your function execution. Learn more about returning results from your function.

```
"handled"
```

Saved Test Event

test

```
{
  "account": "123456789012",
  "region": "us-east-1",
  "detail-type": "CodeDeploy Deployment State-change Notification",
  "source": "aws.codedeploy",
  "version": "0",
  "time": "2016-06-30T22:06:31Z",
  "id": "c071bfbf-83c4-49ca-a6ff-3df053957145",
  "resources": [
    "arn:aws:codedeploy:us-east-1:123456789012:application:myApplication",
    "arn:aws:codedeploy:us-east-1:123456789012:deploymentgroup:myApplication/myDeplo
  ],
  "detail": {
    "instanceGroupId": "9fd2fbef-2157-40d8-91e7-6845af69e2d2",
    "region": "us-east-1",
    "application": "myApplication",
    "deploymentId": "d-123456789",
    "state": "FAILURE",
    "deploymentGroup": "myDeploymentGroup"
  }
}
```

invoke

API Gateway Invocation

stu 3:01 PM

hello again @deploystatus!

DeployStatus APP 3:01 PM

```
|                          :time |        :group |  :app |   :state |          :id |
|--------------------------------+---------------+-------+----------+--------------|
| Thu Jun 14 16:08:10 UTC 2018 | stu-8-compute | stu-8 |   :START | d-FQFA6JPPT |
| Thu Jun 14 16:08:14 UTC 2018 | stu-8-compute | stu-8 |   :START | d-FQFA6JPPT |
| Thu Jun 14 16:08:37 UTC 2018 | stu-8-compute | stu-8 | :SUCCESS | d-FQFA6JPPT |
| Thu Jun 14 16:08:38 UTC 2018 | stu-8-compute | stu-8 |   :START | d-FQFA6JPPT |
| Thu Jun 14 16:09:01 UTC 2018 | stu-8-compute | stu-8 | :SUCCESS | d-FQFA6JPPT |
| Thu Jun 14 16:09:03 UTC 2018 | stu-8-compute | stu-8 | :SUCCESS | d-FQFA6JPPT |
```

invoke

Local Dev

Choose client or in-mem implementation by context

```
1 {:server-type :ion
2  :region "us-east-1"
3  :system "stu-8"
4  :query-group "stu-8"
5  :endpoint “http://entry.stu-8..."
6  :proxy-port 8182}
```

https://github.com/Datomic/ion-event-example/blob/master/src/datomic/ion/event_example.clj

dev

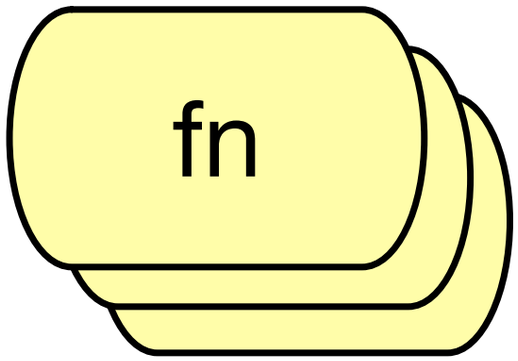
Testing

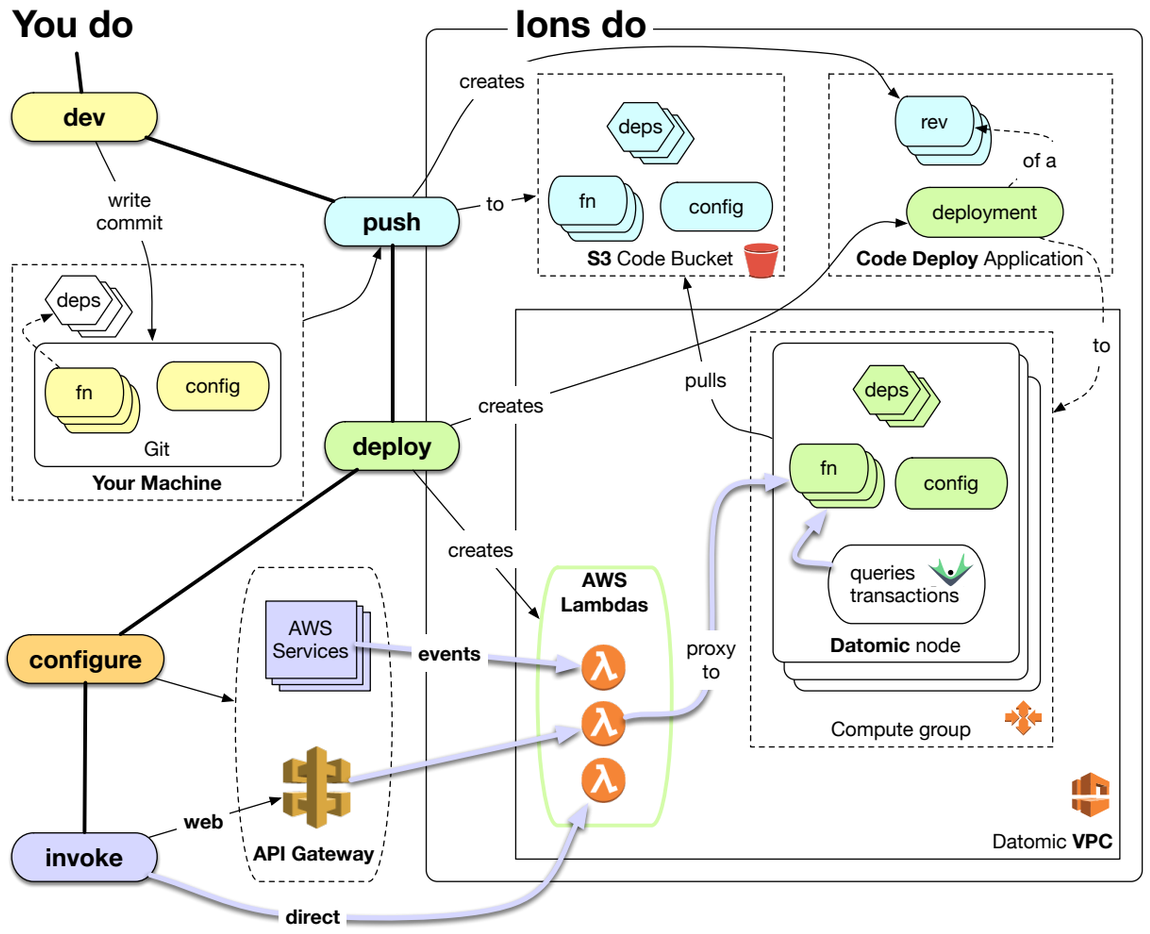
```clojure
(defn sstream [^String s] (java.io.ByteArrayInputStream. (.getBytes s)))
(def conn (ev/get-conn))
(def db (d/db conn))
@(def channel (ev/get-config db :slack/channel))

(ev/post-slack-message channel "hi")

(def cd-failure (slurp "fixtures/events/cd/failure.json"))
(ev/event-handler {:input cd-failure})

(ev/recent-deploys db (ev/hours-ago 24))

(ev/tableize-deploys *1)

(ev/slack-event-handler* {:body (sstream "{}")})

(let [json (format "{\"token\": \"%s\"}"
                   (ev/get-config db :slack/verification-token))]
  (ev/slack-event-handler* {:body (sstream json)}))
```

https://github.com/Datomic/ion-event-example/blob/master/siderail/demo.clj

dev

Focus on the Fn!





For More Info

Sign Up For Datomic on AWS Marketplace

Ions Docs