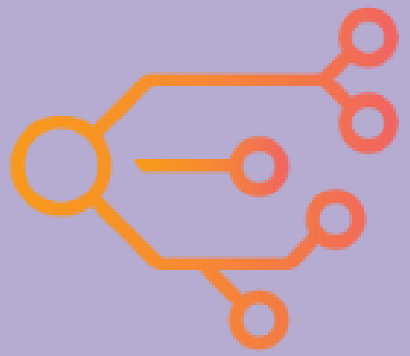
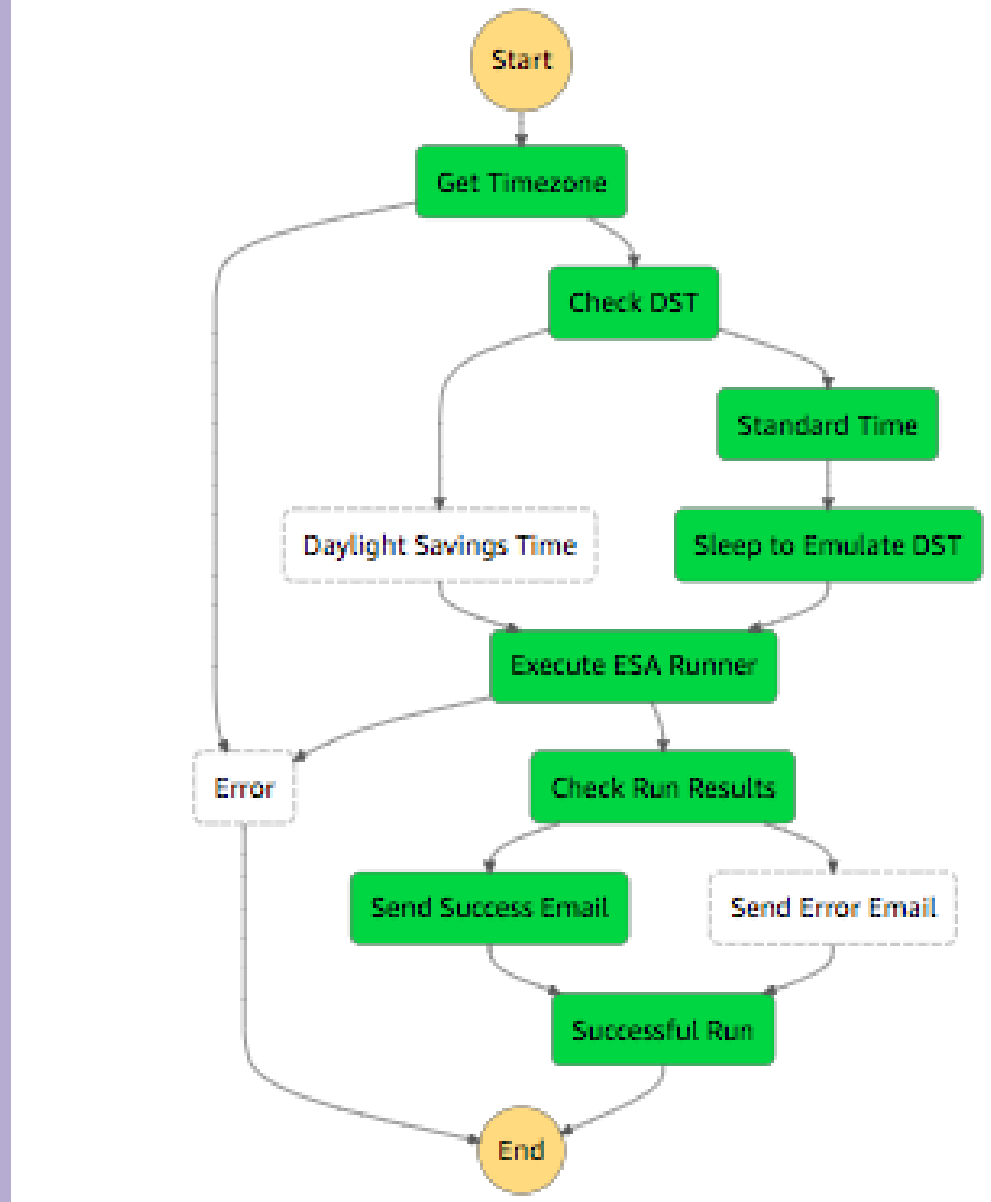


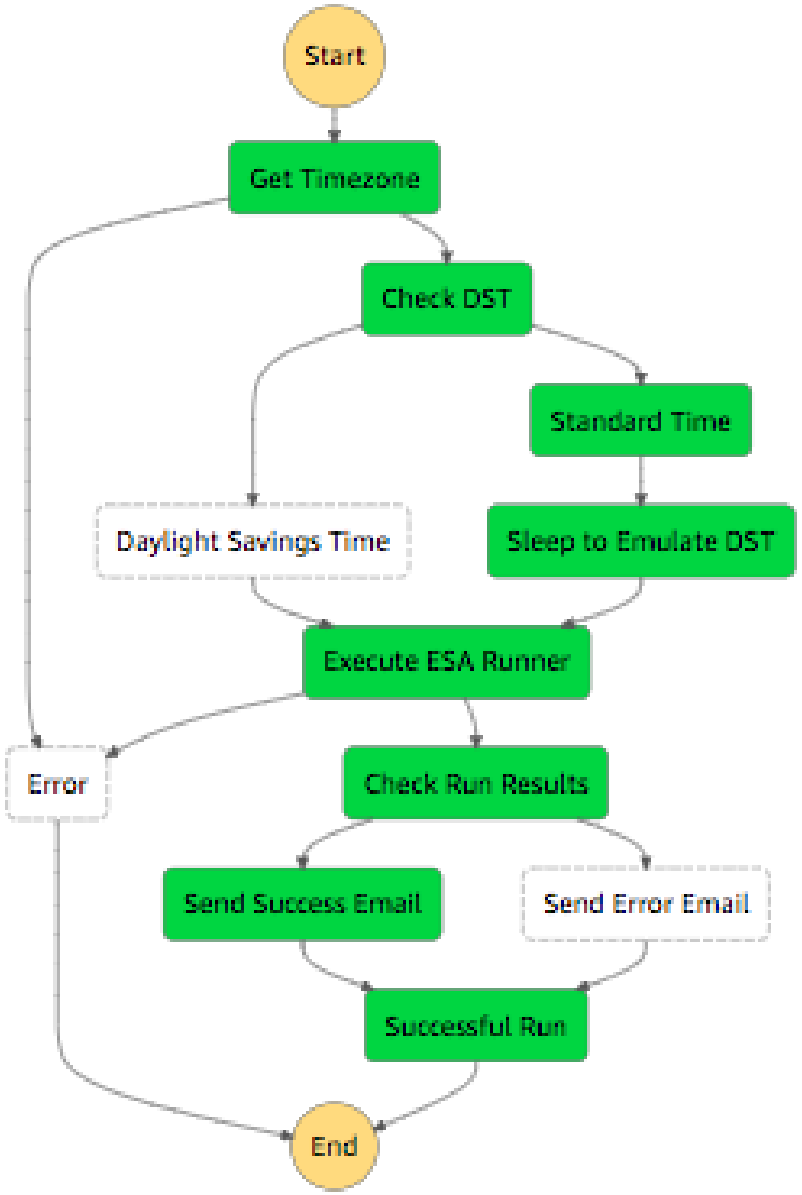
AWS Step Functions

Case Study for Cronjobs in the Cloud

Step function?

- Finite-state machine
- Put data in a box
- Lines between boxes

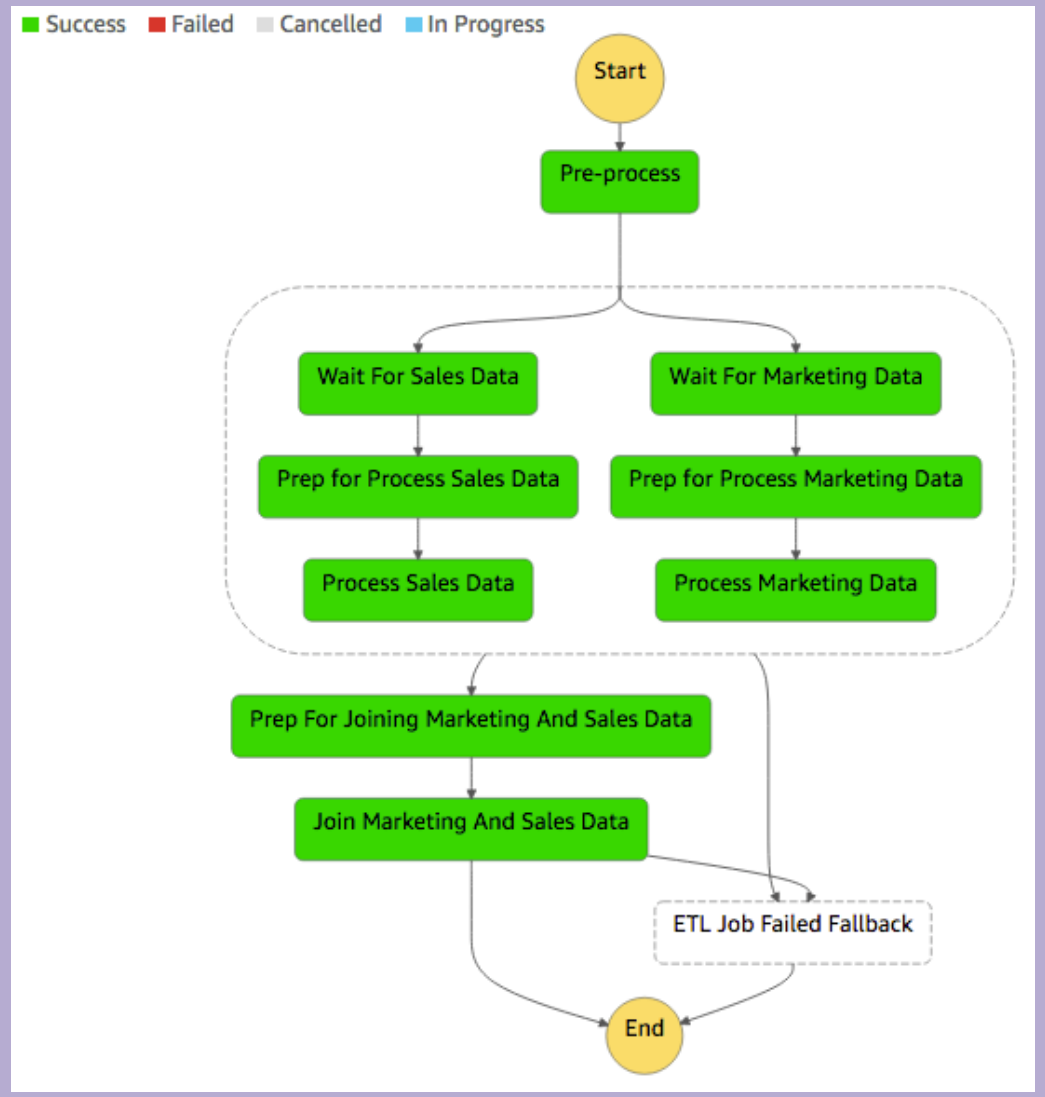




In Progress Succeeded Failed Cancelled Caught Error

What now?

- The boxes do things
  - Lambda
  - SNS / SQS message
  - Start/stop EC2s, ECS tasks, etc
  - Wait for a human to do a thing
  - Sleep



Use cases

- Orchestrate things
- ETL jobs in sequence
- Server patching
- Sky's the limit

Why not just use Lambda?

Lambda and the AWS SDK could be used to achieve anything that step functions can.

But step functions make certain things easier.

Like babysitting & re-trying failed Lambdas. No 15 minute limit here 😄

The Case Study

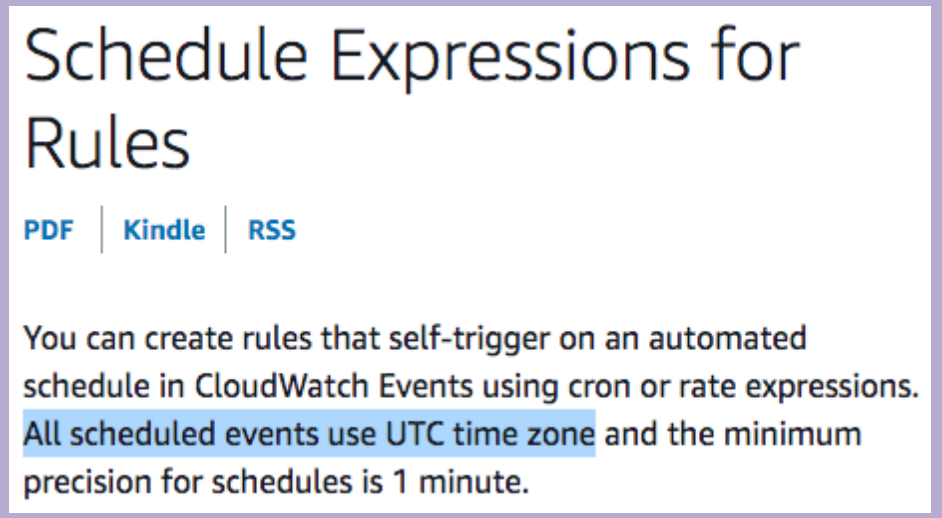
- Vendor app
- Vendor wrote a custom ASP page that does an ETL from MyHR
- Human goes to the page every Monday morning
- Probably takes a screenshot of the resulting import stats
- **Yuck.** Let's automate that!

Development

1. Write a Lambda function
    1. Load some SSM params w/ URLs & credentials
    2. Do an SSO login
    3. `GET` the ASP page
    4. Parse the HTML to extract results
    5. Use MJML to make a nice responsive email w/ the results
    6. Send to a distro list w/ Simple Email Service
2. Terraform the infrastructure
3. Test Lambda execution

Schedule in CloudWatch

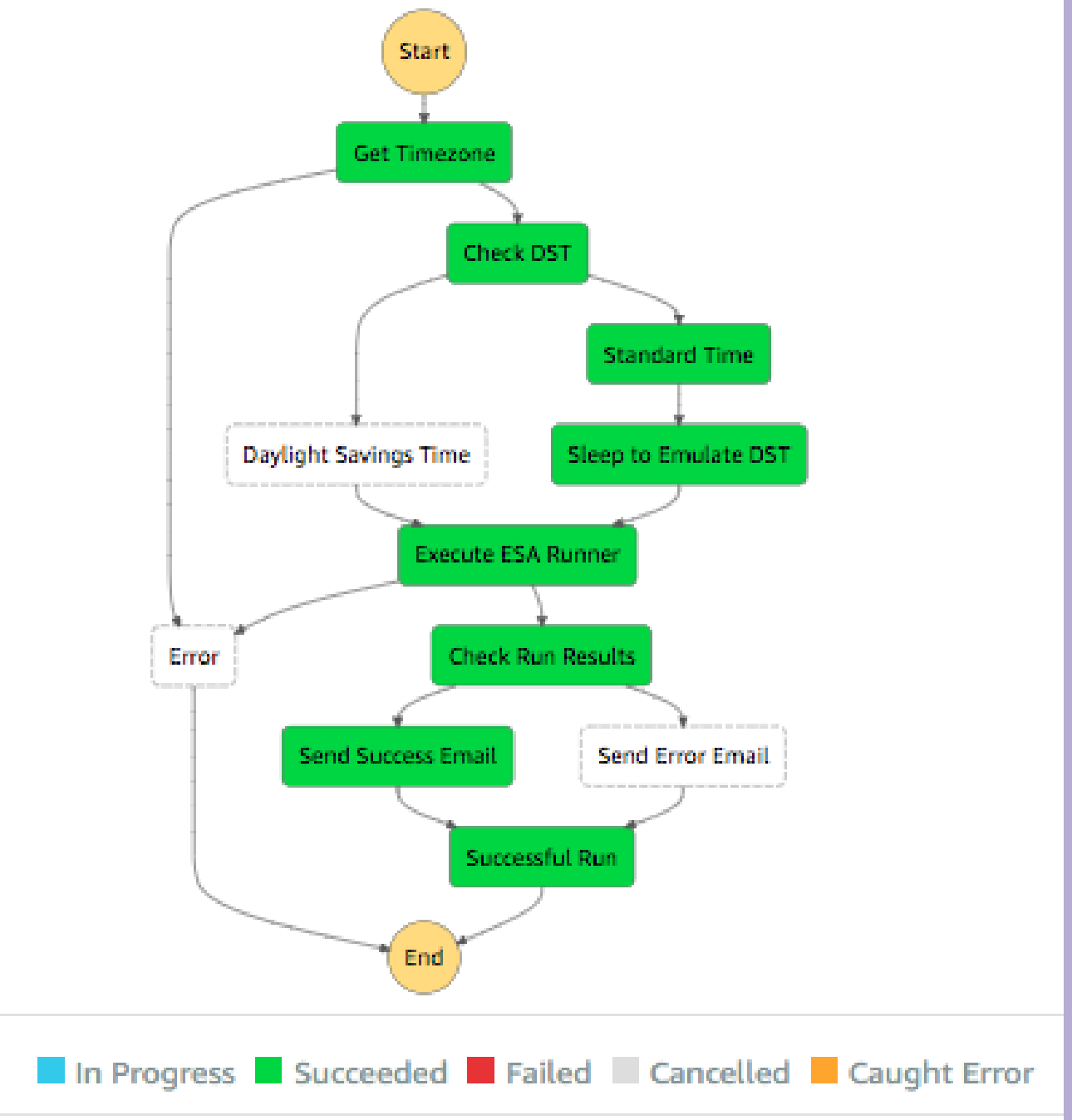
> Hi Nick,
>
> Please make sure this job runs EXACTLY at eight AM on Mondays.
>
> Regards,
> The Business




-Hmm
"oh no"

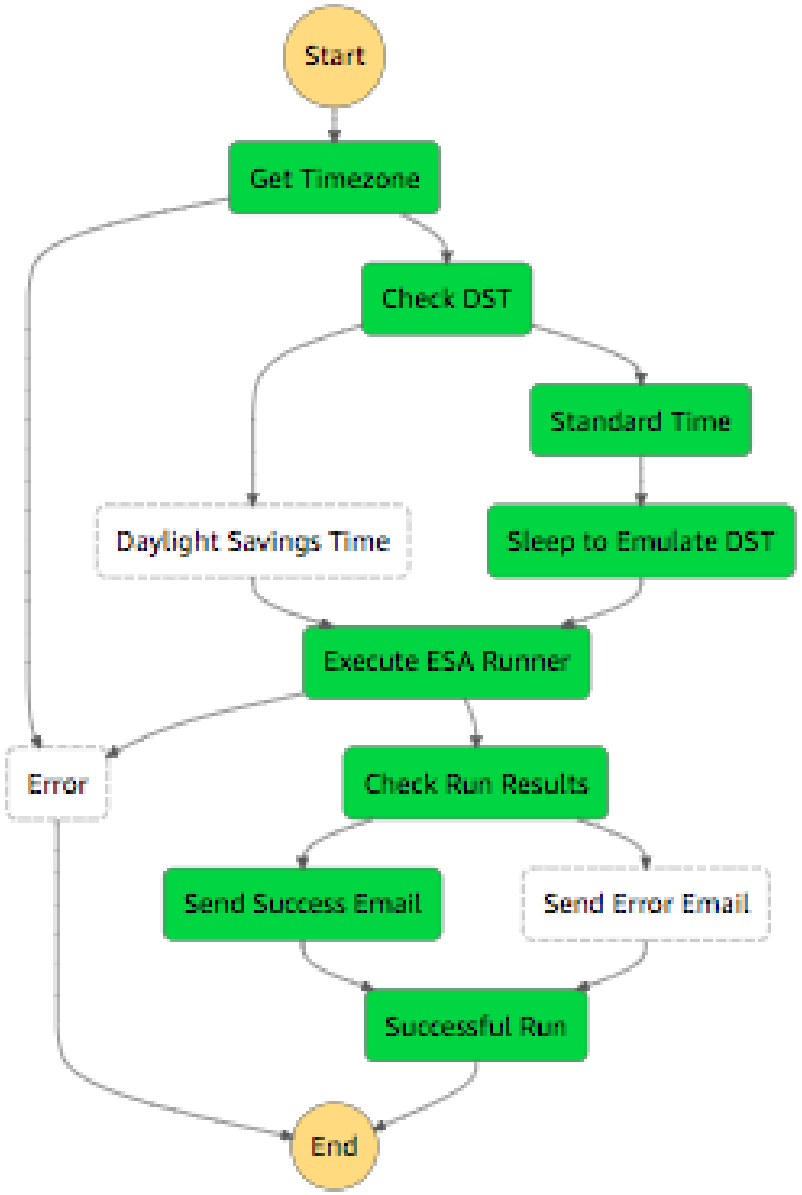
Scheduling DST-aware CloudWatch Event Rules

- Not a thing.
- Lambda sleeps for an hour?
  - Nope: 15 minute maximum execution time
- Adjust CloudWatch rule twice a year?
  - lol



Step Functions to the Rescue

- Step Functions have a "wait" state that costs nothing ($$)
- CloudWatch fires the Step Function at the early hour
  - Detects if it needs to sleep



In Progress Succeeded Failed Cancelled Caught Error

Timezone

```
const moment = require('moment-timezone');

exports.timezone = (event) => {
    return {
        "Timezone": moment()
            .tz('America/Chicago')
            .format('z')
    }
}
```

Step Function Choice

```
"Check DST": {
    "Type": "Choice",
    "Choices": [{
            "Not": {
                "Variable": "$.Payload.Timezone",
                "StringEquals": "CST"
            },
            "Next": "Daylight Savings Time"
        },
        {
            "Variable": "$.Payload.Timezone",
            "StringEquals": "CST",
            "Next": "Standard Time"
        }
    ]
},
```

Refining It

- While I'm using this nifty Lambda orchestration tool, I can refactor my code to be more focused.
  - "Main" lambda logs in, hits the page, parses HTML
  - Passes the "job log" out
  - Step function choice: >0 job log entries = success Lambda
  - Otherwise failure Lambda
- New lambdas do the emailing parts w/ MJML & SES

"Hm, what if a Lambda fails?"

```
resource "aws_cloudwatch_metric_alarm" "failed_executions_alarm" {
  alarm_name          = "${var.lambda_name}-${var.env}-ExecutionsFailed"
  comparison_operator = "GreaterThanThreshold"
  evaluation_periods  = "1"
  metric_name         = "ExecutionsFailed"
  namespace           = "AWS/States"
  period              = "120"
  statistic           = "Sum"
  threshold           = "0"
  alarm_description   = "Monitors for ${var.app_name} ${var.env} step function errors."
  treat_missing_data  = "ignore"
  actions_enabled     = "${var.alert_on_failure}"
  alarm_actions       = ["${local.opsgenie_sns_arn}"]

  dimensions = {
    StateMachineArn = "${aws_sfn_state_machine.runner_step_function.id}"
  }
}
```

Costs?

- AWS Step Functions are billed per state transition.
  - Lambda exec time, SES costs, etc all still applicable
- 4,000 per month are free
  - $0.025 / 1,000 after that
- So, not really expensive.

Next Steps

The IaC module I wrote for this is more generally applicable -- do a thing at the right time, decide if it succeeded, and then call a success or failure Lambda. If a technical problem blows up the state machine, cry to the on-call person.

I plan to turn this into a sharable terraform module. Bring your own Lambda implementation, SNS ARN, and CloudWatch schedule 😄

The End

Thanks for listening everybody!

- https://github.com/nie7321/step-function-talk
- https://mjml.io/ - responsive email templates

If you have any questions, you can find me in the DevIT slack's `#laravel` channel!