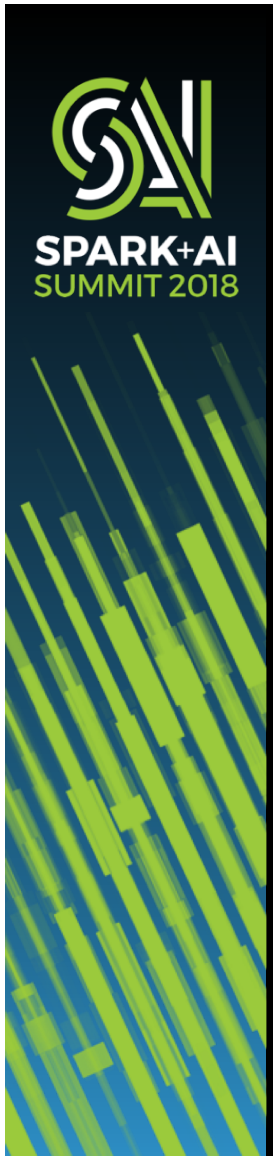


HIPAA Compliant Deployment of Apache Spark on AWS

Nitin Panjwani, Christian Nuss
Collective Health

#Ent8SAIS

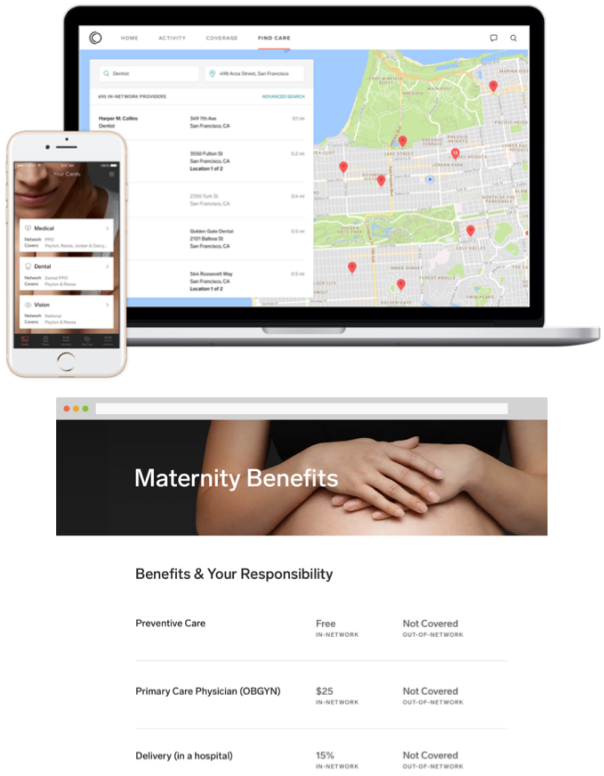
About Collective Health

What do we do

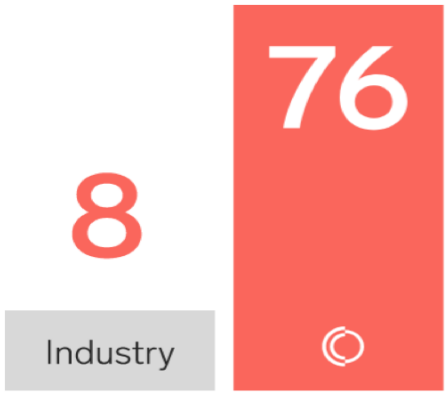
Collective Health provides a platform for employers to provide healthcare benefits to their workforce.

Objective

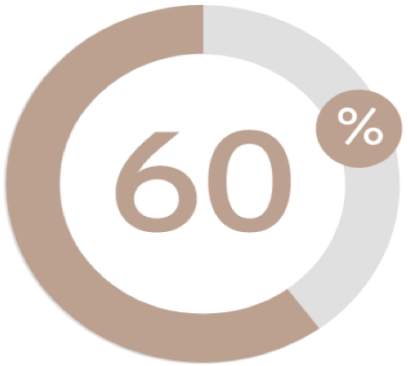
Data Driven approach to reduce the cost for the employer and provide a high quality service to members.



About Collective Health

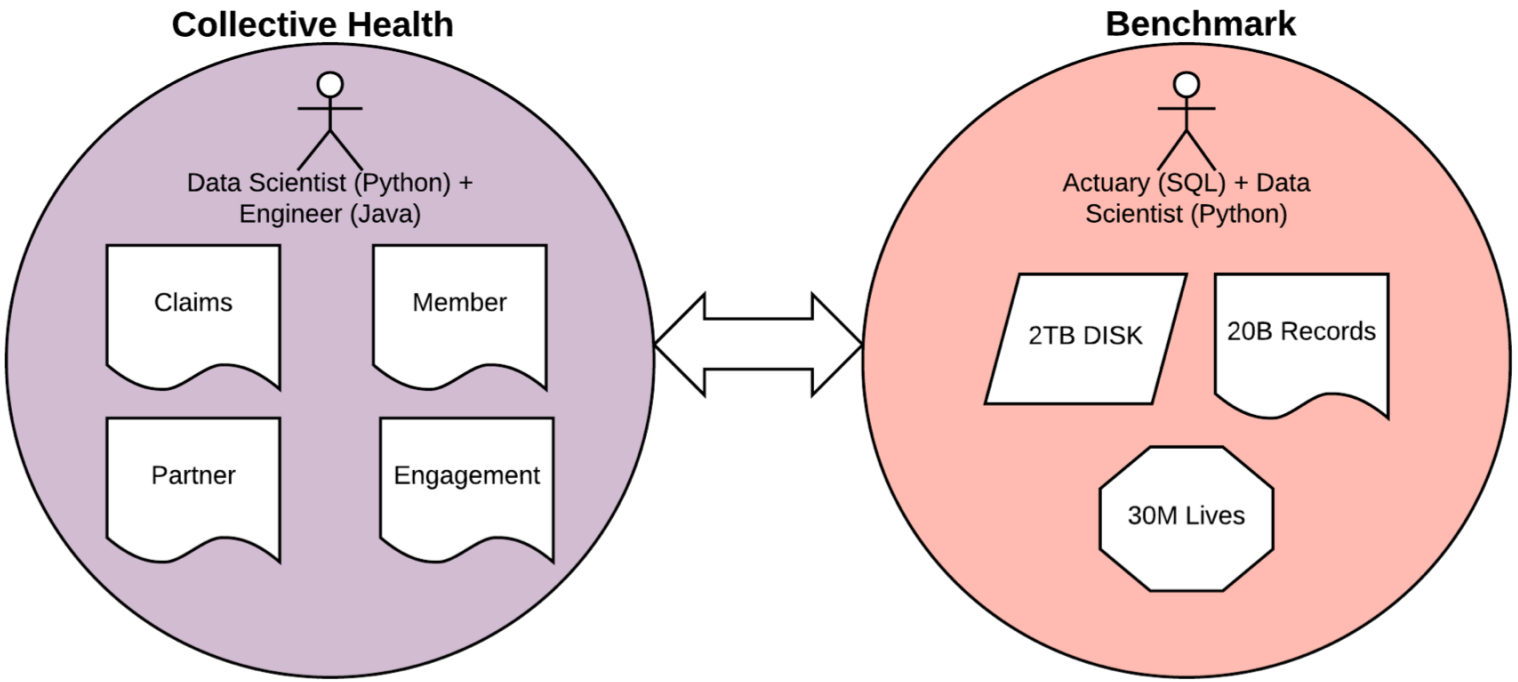


Our Net Promoter Score is nearly 10x the industry average



Clients have reported a 60% reduction in employee questions

Data Footprints



Analytics Platform Requirements

- Diverse user backgrounds: SQL, Python, Java
- Big Benchmark Data
- Shared Notebooks (Zeppelin) for Big Data
- Advanced ML models
  - Risk scores: regression models
  - Classifiers to find members on our proprietary recommendation engine

Our Compliance Landscape

HIPAA and PHI

- **H**ealth **I**nsurance **P**ortability and **A**ccountability **A**ct

Safeguards

- **P**rotected **H**ealth **I**nformation (PHI)

SOC 2

- **S**ervice **O**rganization **C**ontrol Type **2**

Ensures

- Privacy
- Security
- Availability
- Integrity
- Confidentiality

Our Requirements

HIPAA and SOC 2

- Encryption
- Authentication
- Authorization
- Identity
- Access Logging
- Auditability
- Scalability
- Repeatability

Data Science Challenges

- Access to data is not easily available due to regulatory compliance
- Systems in cloud should have access to data
- We need Cloud version of Jupyter/Zeppelin
- Bringing in new technologies is not easy
  - Framework should be compliant
- Data footprint should be auditable

What is Amazon EMR?

Elastic **M**ap**R**educe

"Amazon EMR provides a **managed** Hadoop framework that makes it easy, fast, and cost-effective to process vast amounts of data across dynamically scalable **Amazon EC2 instances**."

"You can also run other **popular** distributed frameworks such as **Apache Spark**, HBase, Presto, and Flink."

What does EMR do?

Infrastructure Management

Software Installation

Configuration Management

Monitoring & Administration

EMR Services

Flink	Ganglia	Hadoop	HBase	HCatalog
Hive	Hue	Livy	Mahout	MXNet
Oozie	Phoenix	Pig	Presto	Spark
Sqoop	Tez	Zeppelin	ZooKeeper	

EMR Services

Flink	Ganglia	Hadoop	HBase	HCatalog
Hive	Hue	Livy	Mahout	MXNet
Oozie	Phoenix	Pig	Presto	Spark
Sqoop	Tez	Zeppelin	ZooKeeper	

Customizing EMR

General Settings

Bootstrap Actions

Configurations

Start-Up Steps

Customizing EMR

General Settings

- Security Groups
- IAM Roles
- Logging
- Encryption
- Monitoring
- Auto-Scaling

Customizing EMR

Bootstrap Actions

- Shell scripts in S3
- Runs on all nodes
- Runs prior to EMR installs

Examples:

- Set up SSO on Zeppelin
- Additional Python or Java packages
- Custom monitoring

Customizing EMR

Configurations

- JSON Format
- Overrides defaults
- Difficult to understand and verify

Examples:

- Spark Defaults
- Yarn Container Limits
- Add TLS to Zeppelin Web UI
- Verbose Logging

https://docs.aws.amazon.com/emr/latest/ReleaseGuide/emr-release-5x.html

Customizing EMR

Start-Up Steps

- Hadoop JAR
- AWS "script-runner.jar"
- Runs any job once cluster is ready

Examples:

- Test connectivity
- Set Zeppelin Interpreter settings
- Run any arbitrary script against a running cluster

PSA: Ephemeral Infrastructure

- Most changes require the cluster to recreate
- Treat EMR clusters as immutable
- HDFS data is not durable
- Cluster changes take 5-10 minutes

EMR Customizations

Complex and Fragmented

General Settings	Bootstrap Actions	Configurations	Start-Up Steps
UI or API	Script	JSON	Hadoop JAR

Our Solution

Amazon
EMR

HashiCorp
Terraform

What is Terraform?

Infrastructure As Code | State | Reproducible Infrastructure

What is Terraform?

Declarative Configuration Files

```
resource "aws_s3_bucket" "bucket" {
  bucket = "${var.cluster_name}-${terraform.env}"
  acl    = "private"
}
```

```
resource "aws_emr_cluster" "cluster" {
  name           = "${var.cluster_name}"
  release_label  = "emr-5.13.0"
  applications   = ["Spark", "Zeppelin", ...]
  …
  log_uri =
          "s3n://${aws_s3_bucket.bucket.id}/logs"
  …
  master_instance_type    = "c4.large"
  core_instance_type      = "c4.large"
  core_instance_count     = 3
}
```

What is Terraform?

API Calls, Codified!

```
$> terraform apply

aws_s3_bucket.bucket: Creating...
  …
  bucket:                   "" => "cnuss-test-us-west-1-default"
  …
aws_s3_bucket.bucket: Creation complete after 3s (ID: cnuss-test-us-west-1-default)
aws_emr_cluster.cluster: Creating...
…
aws_emr_cluster.cluster: Creation complete after 4m23s (ID: j-3CUR3J8Y6NYE9)


Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

https://github.com/hashicorp/terraform

What is Terraform?

Changes, As Code!

```
resource "aws_s3_bucket" "bucket" {
  bucket = "${var.cluster_name}-${terraform.env}"
  acl    = "private"

  versioning {
    enabled = true
  }
}
```

```
resource "aws_emr_cluster" "cluster" {
  name          = "${var.cluster_name}"
  release_label = "emr-5.13.0"
  applications  = ["Spark", "Zeppelin", ...]
  …
  log_uri =
          "s3n://${aws_s3_bucket.bucket.id}/logs"
  …
  master_instance_type    = "c4.large"
  core_instance_type      = "c4.large"
  core_instance_count     = 3
}
```

What is Terraform?

Stateful

```
$> terraform apply

aws_s3_bucket.bucket: Refreshing state... (ID: cnuss-test-us-west-1-default)
aws_emr_cluster.cluster: Refreshing state... (ID: j-3CUR3J8Y6NYE9)

aws_s3_bucket.bucket: Modifying... (ID: cnuss-test-us-west-1-default)
  versioning.0.enabled: "false" => "true"
aws_s3_bucket.bucket: Modifications complete after 2s (ID: cnuss-test-us-west-1-default)

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

https://github.com/hashicorp/terraform

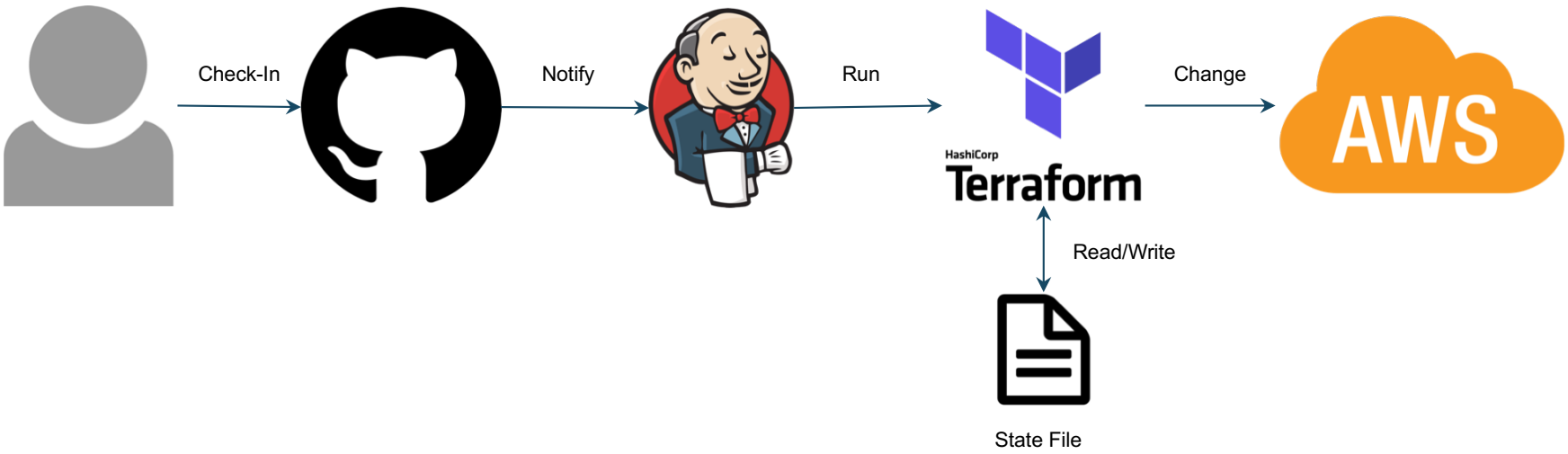
What is Terraform?

Also, dangerous…

```
$> terraform destroy

Terraform will perform the following actions:
  - aws_emr_cluster.cluster
  - aws_s3_bucket.bucket

aws_emr_cluster.cluster: Destroying... (ID: j-3CUR3J8Y6NYE9)
aws_emr_cluster.cluster: Destruction complete after 1m48s (ID: j-3CUR3J8Y6NYE9)
aws_s3_bucket.bucket: Destroying... (ID: cnuss-test-us-west-1-default)
aws_s3_bucket.bucket: Destruction complete after 0s

Destroy complete! Resources: 1 destroyed.
```

https://github.com/hashicorp/terraform

Typical Terraform Workflow



EMR + Terraform

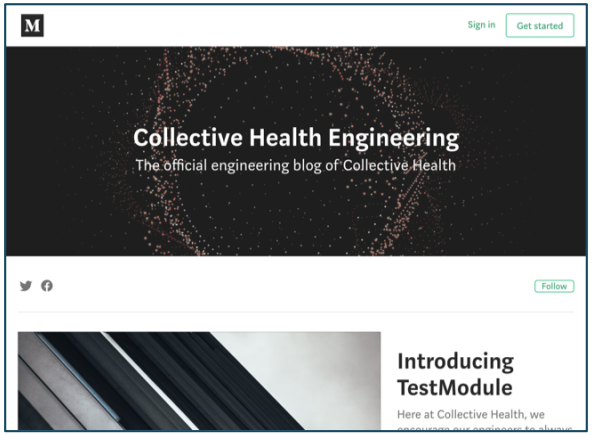
Our Customizations

- TLS Everywhere (HDFS, Zeppelin)
- All Logs to S3 for Log Analysis
- S3 Encryption and Versioning
- Zeppelin + SSO + 2FA
- Custom Monitoring Agents Installed
- Autoscaling
- Terraform Code in GitHub, Jenkins runs Terraform

Next Steps

- Identity and logging of `spark-submit` commands
- Expose `spark-submit` to other applications and engineers
- Jupyter Access to Spark
- Use EMRFS instead of HDFS (waiting on Terraform)
- Additional Monitoring and Autoscaling

Sample Code / Starter Kit



https://eng.collectivehealth.com

https://twitter.com/hashtag/Ent8SAIS

https://linkedin.com/in/christiannuss

Key Takeaways

- Big Data Analytics in Healthcare is very challenging
- EMR "out-of-the-box" requires extensive customizations to comply with regulations
  - Need to add encryption, logging, identity
- Terraform(API for infra) adds value
  - Simplifies confusing configuration options
  - Repeatability
  - All configuration is code

Our Results So Far...

- Developed demographic factors
  - Variation in average per capita cost due to age and gender
- Geo cost variation factors based on MSA
  - Variations in the average per capita cost due to cost and use of medical practice by metropolitan statistical areas (MSAs)
- Industry factors
  - Variations due to industry
- Cost variations by clinical conditions
- Disease prevalence
- Risk Scores (regression model)

Thank You!

Q&A?

Nitin Panjwani
Principal Data Scientist
Collective Health
linkedin.com/in/npanj

https://collectivehealth.com/jobs

Christian Nuss
Senior SRE
Collective Health
github.com/cnuss