amazon web services

Masterclass

Amazon Redshift

Ian Massingham — Technical Evangelist

ianmas@amazon.com

@IanMmmm

intel

Masterclass

1. A technical deep dive that goes beyond the basics
2. Intended to educate you on how to get the best from AWS services
3. Show you how things work and how to get things done

Amazon Redshift

A fast, fully managed, petabyte-scale data warehouse

Makes it simple & cost-effective to analyse all your data

You can continue to use your existing tools

Start small & scale to a petabyte or more

Costs less than $1,000/terabyte/year

Scalable

Optimised for
Data Warehousing

No Up-Front Costs

Amazon Redshift

Compatible

Simple

Secure

AWS Official Blog

Amazon Redshift – The New AWS Data Warehouse

by Jeff Barr | on 28 NOV 2012 | Permalink | Comments

You may have noticed that we announced Amazon Redshift today a data warehouse service that is a lot faster, simpler, and less expensive than alternatives available today.

Let's Talk About Data

A data warehouse is a specialized type of relational database, optimized for high-performance analysis and reporting of transactions. It collects current and historical transactional data from many disparate operational systems (manufacturing, finance, sales, shipping, etc.) and pulls it together in one place to guide analysis and decision-making. Without a data warehouse, a business lacks an integrated view of its data. Essential reports aren't available in time, visibility into changes that affect the business is reduced, and business suffers.

To date, setting up, running, and scaling a data warehouse has been complicated and expensive. Maintaining and monitoring all the moving parts (hardware, software, networking, storage) requires deep expertise across a number of disciplines. At scale, even comparatively simple things like loading or backing up data get hard. Getting fast performance requires mastery of complicated query plans, access methods, and indexing techniques. No surprise, traditional data warehouses have been seen as gold-plated, premium-priced products with substantial upfront costs.

Enter Amazon Redshift

Amazon Redshift makes it easy for you setup, run, and scale a data warehouse of your own. AWS customers such as Netflix, JPL, and Flipboard have been testing it as part of a private beta. We are launching Amazon Redshift in a limited public beta mode today.

Amazon Redshift is a massively parallel, fully-managed data warehouse service, designed for data sets from hundreds of gigabytes to several petabytes in size, and appropriate for an organization of any size from a startup to a multi-national at a price point that will take you by surprise. Amazon Redshift is fully managed, so you no longer need to worry about provisioning hardware, installation, configuration or patching of system or database software. Because your business is dependent on your data, Amazon Redshift takes care to protect it by replicating all data within the cluster as well as in S3.

Amazon Redshift won't break the bank (or your credit card, since it is completely pay-as-you-go). We did the math and found that it would generally cost you between $19,000 and $25,000 *per terabyte per year* at list prices to build and run a good-sized data warehouse on your own. **Amazon Redshift, all-in, will cost you less than $1,000 per terabyte per year.** For that price you get all of the benefits that I listed above without any of the operational headaches associated with building and running your own data warehouse.

Like every AWS service, you can create and manipulate an Amazon Redshift cluster using a set of web service

https://aws.amazon.com/blogs/aws/category/amazon-redshift/

User Defined Functions for Amazon Redshift

by Jeff Barr | on 11 SEP 2015 | in Amazon Redshift | Permalink | Comments

The Amazon Redshift team is on a tear. They are listening to customer feedback and rolling out new features all the time! Below you will find an announcement of another powerful and highly anticipated new feature.

— Jeff;

Amazon Redshift makes it easy to launch a petabyte-scale data warehouse. For less than $1,000/Terabyte/year, you can focus on your analytics, while Amazon Redshift manages the infrastructure for you. Amazon Redshift's price and performance has allowed customers to unlock diverse analytical use cases to help them understand their business. As you can see from blog posts by Yelp, Amplitude and Cake, our customers are constantly pushing the boundaries of what's possible with data warehousing at scale.

To extend Amazon Redshift's capabilities even further and make it easier for our customers to drive new insights, I am happy to announce that Amazon Redshift has added scalar user-defined functions (UDFs). Using PostgreSQL syntax, you can now create scalar functions in Python 2.7 custom-built for your use case, and execute them in parallel across your cluster.

Here's a template that you can use to create your own functions:

```sql
CREATE [ OR REPLACE ] FUNCTION f_function_name
( [ argument_name arg_type, ... ] )
RETURNS data_type
{ VOLATILE | STABLE | IMMUTABLE }
AS $$
  python_program
$$ LANGUAGE plpythonu;
```

Scalar UDFs return a single result value for each input value, similar to built-in scalar functions such as `ROUND` and `SUBSTRING`. Once defined, you can use UDFs in any SQL statement, just as you would use our built-in functions.

In addition to creating your own functions, you can take advantage of thousands of functions available through Python libraries to perform operations not easily expressed in SQL. You can even add custom libraries directly from S3 and the web. Out of the box, Amazon Redshift UDFs come integrated with the Python Standard Library and a number of other libraries, including:

- NumPy and SciPy, which provide mathematical tools you can use to create multi-dimensional objects, do matrix operations, build optimization algorithms, and run statistical analyses.

https://aws.amazon.com/blogs/aws/category/amazon-redshift/

Agenda

Why Run Your Data Warehouse on AWS?

Getting Started

Table Design

Data Loading

Working with Data

Backup and Restoration

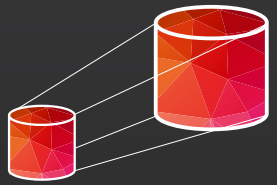
Upgrading & Scaling

WHY RUN YOUR DATA WAREHOUSE ON AWS?

Database Management Challenges

Expensive

Difficult to administer

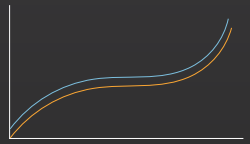


Challenging to scale

Very difficult to switch

Customers asked for data warehousing the AWS way

Easy to provision & scale up massively



No upfront costs, pay as you go

Really fast performance at a really low price

Open and flexible with support for popular BI tools

Amazon Redshift parallelises & distributes everything

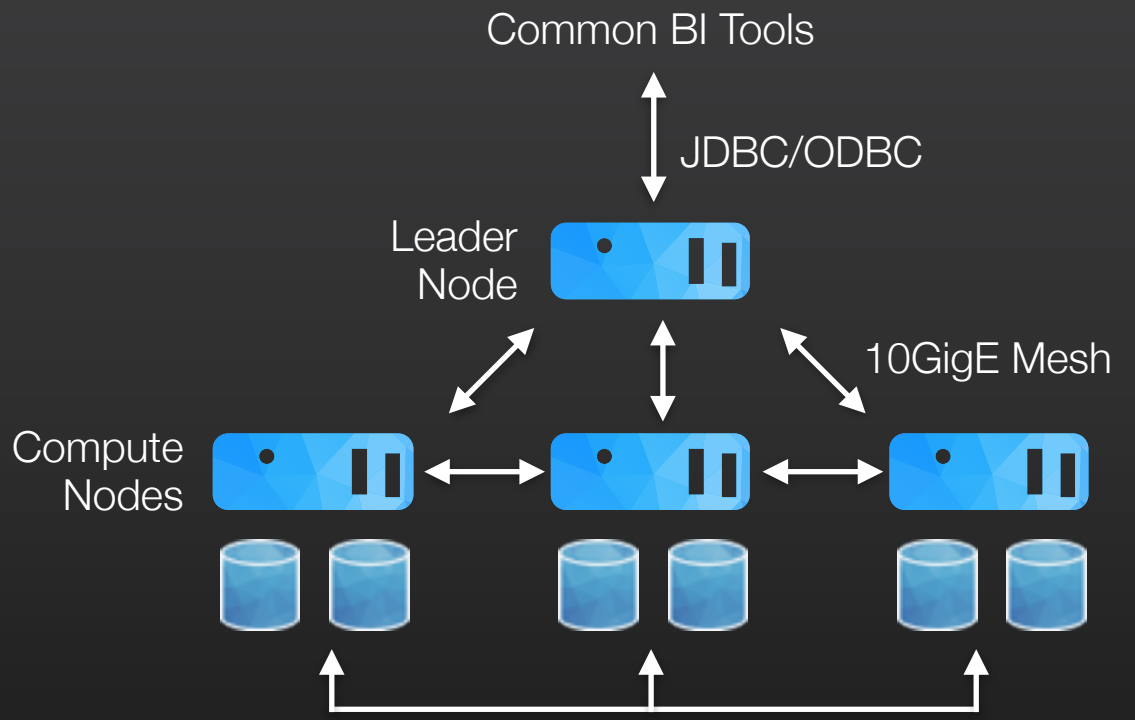
Query

Load

Backup

Restore

Resize



Amazon Redshift lets you start small & grow big

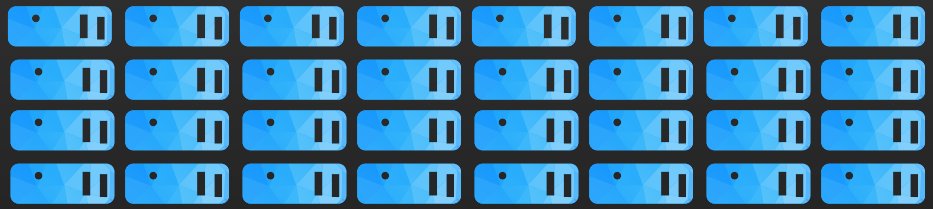
Dense Compute Nodes - dc1.large & dc1.8xlarge

dc1.large nodes

15GiB RAM
2 virtual cores, 10GigE

Single Node (160GB SSD)

Cluster 2-32 Nodes (320GB – 5.12TB SSD)



dc1.8xlarge nodes

244GiB RAM
32 virtual cores, 10GigE

Cluster 2-128 Nodes (up to 326TB SSD)

Amazon Redshift lets you start small & grow big

Dense Storage Nodes - ds1.xlarge, ds1.8xlarge, ds2.xlarge & ds2.8xlarge

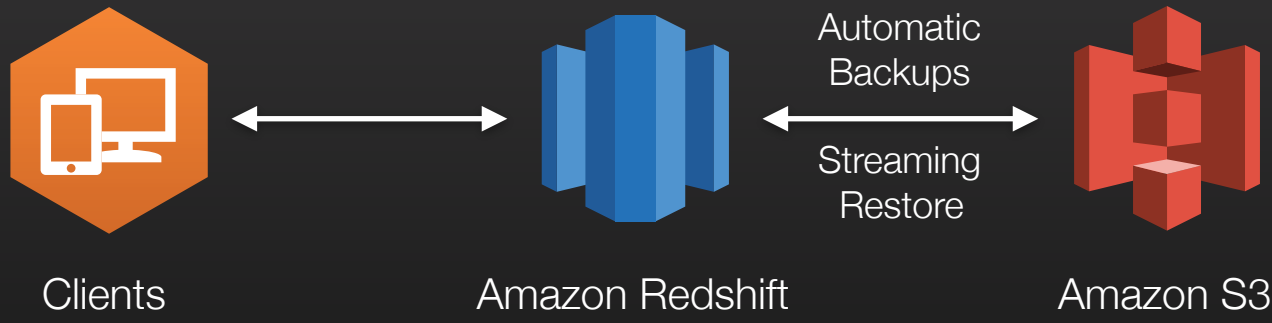
ds2.8xlarge nodes

244GiB RAM
36 virtual cores, 10GigE, 16TB magnetic HDD per node

Cluster 2-128 Nodes (up to 2PB magnetic HDD)

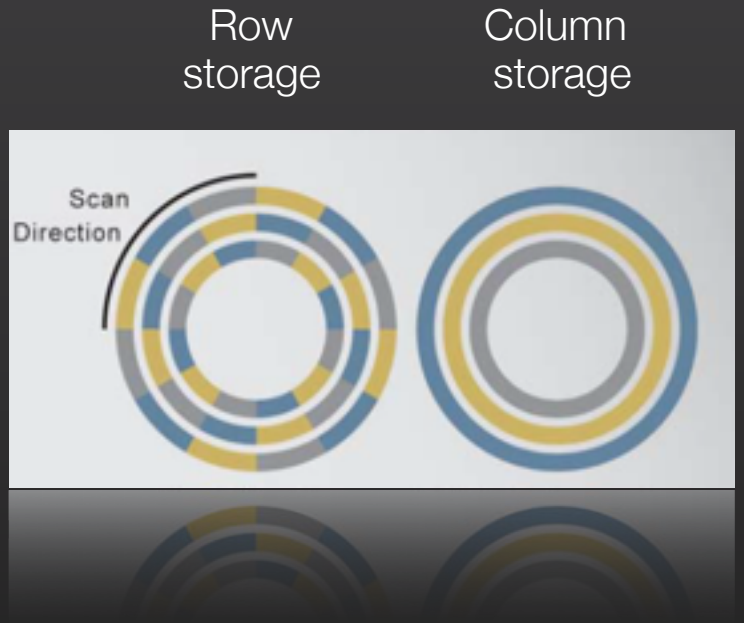
Simplify Cluster Operations with Amazon Redshift

- Built-in security in transit, at rest, when backed up
- Backup to Amazon S3 is continuous, incremental & automatic
- Streaming restores let you resume querying faster
- Disk failures are transparent & nodes recover automatically



Amazon Redshift Dramatically Reduces IO

- Column storage
- Data compression
- Zone maps
- Direct-attached storage
- Large data block sizes



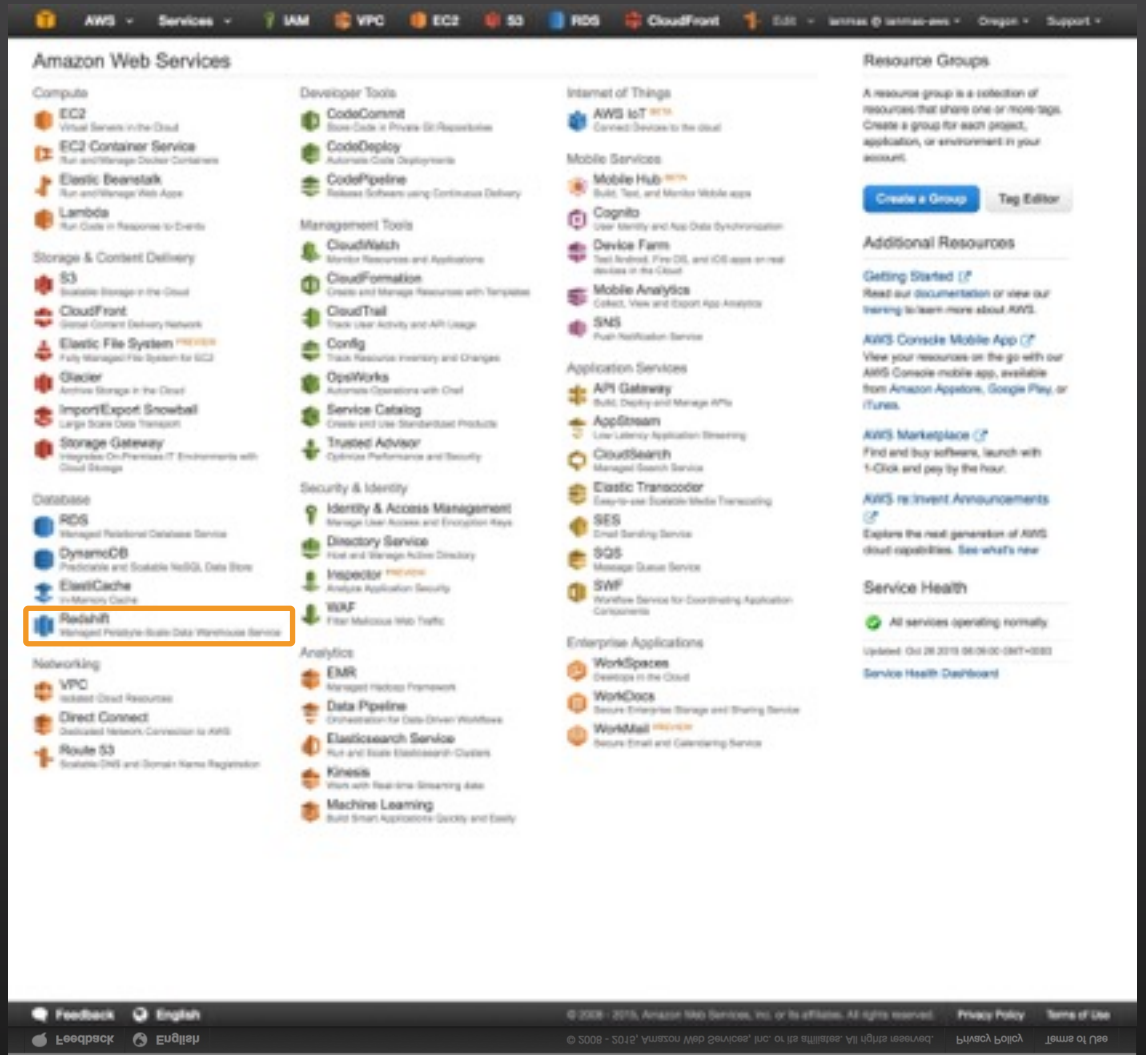
Amazon Redshift in action

FT FINANCIAL TIMES

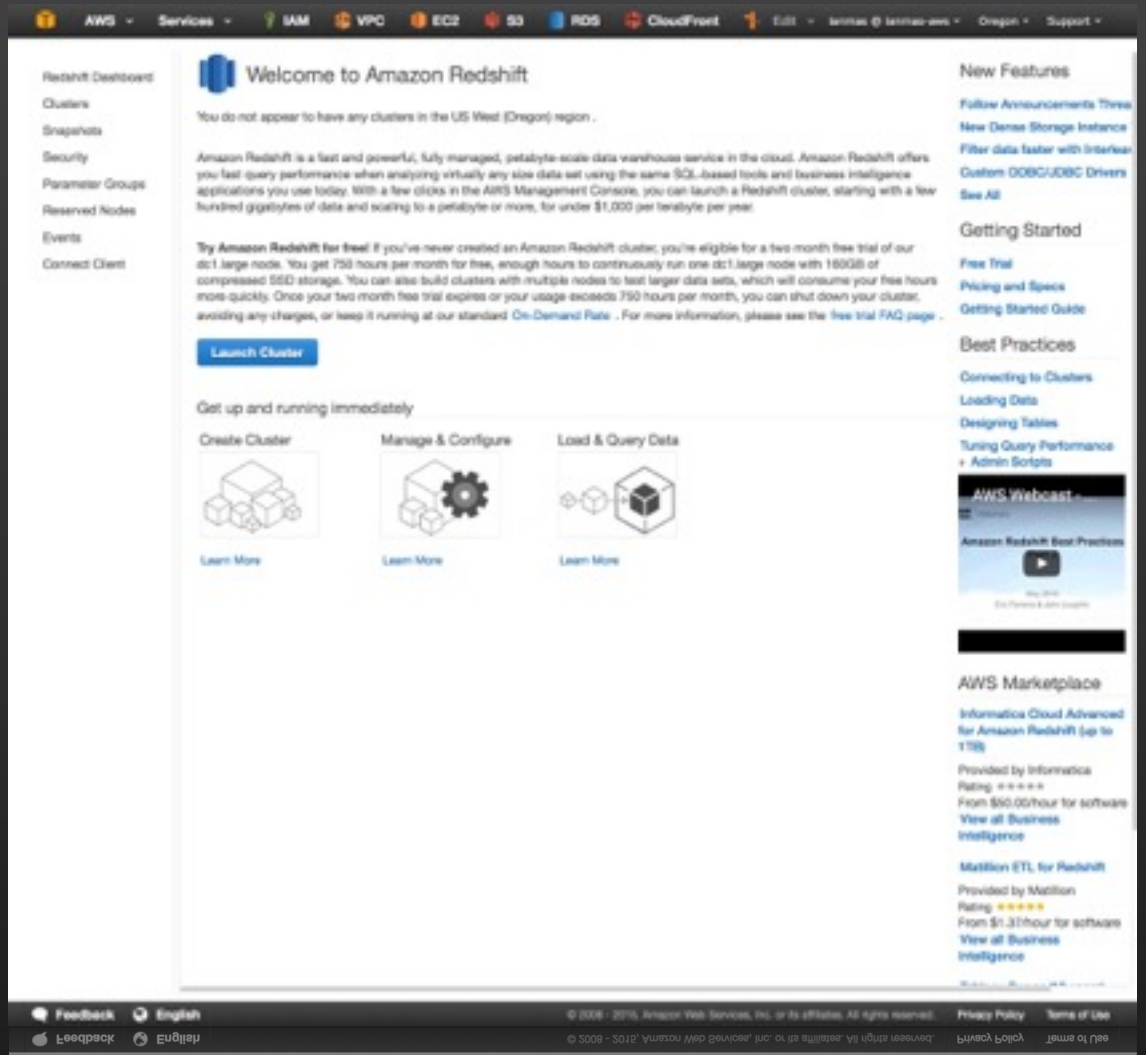
- Needed a way to increase speed, performance and flexibility of data analysis at a low cost
- Using AWS enabled FT to run queries 98% faster than previously—helping FT make business decisions quickly
- Easier to track and analyze trends
- Reduced infrastructure costs by 80% over traditional data center model

GETTING STARTED

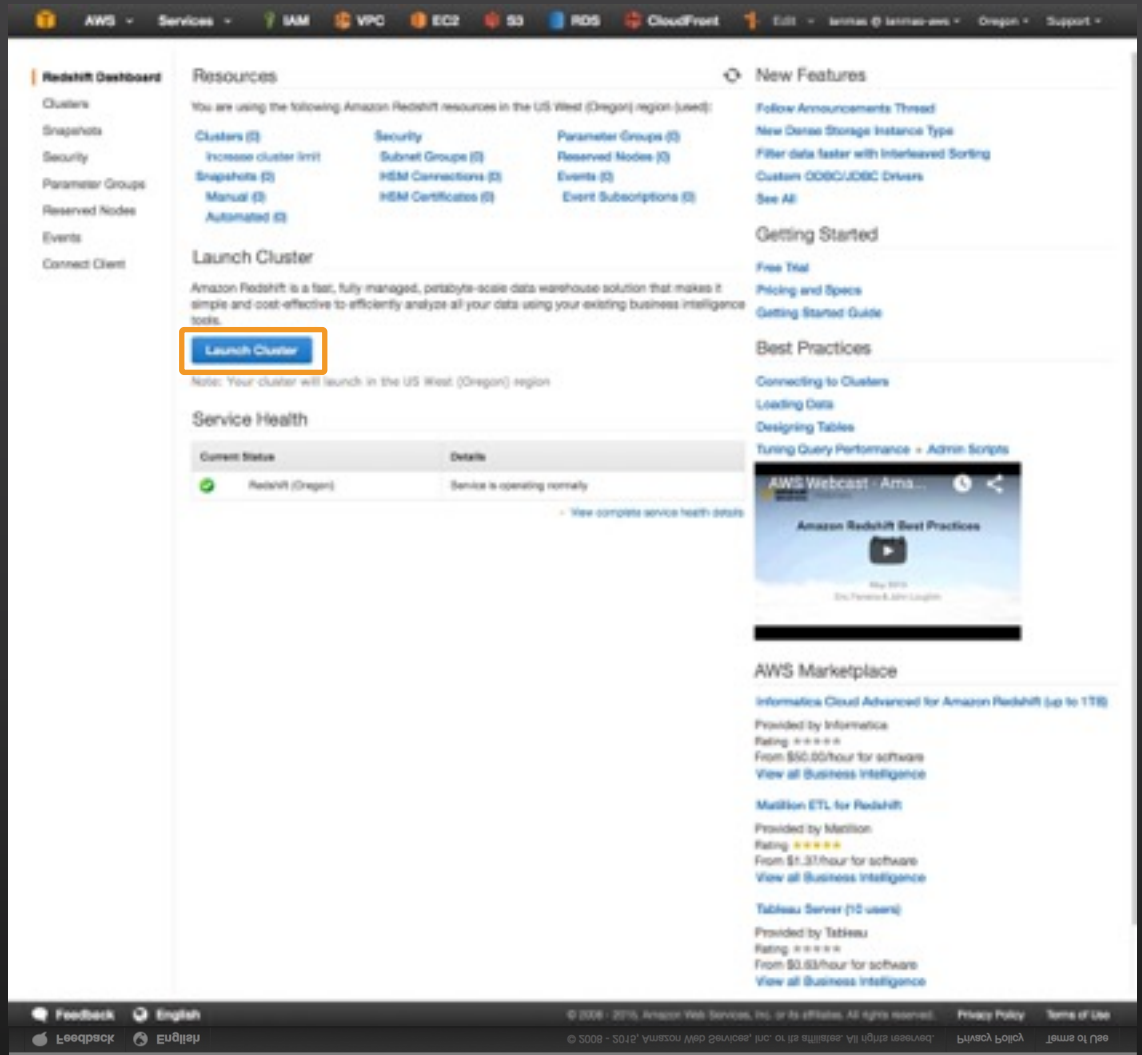
Creating an Amazon Redshift Data Warehouse via the AWS Console



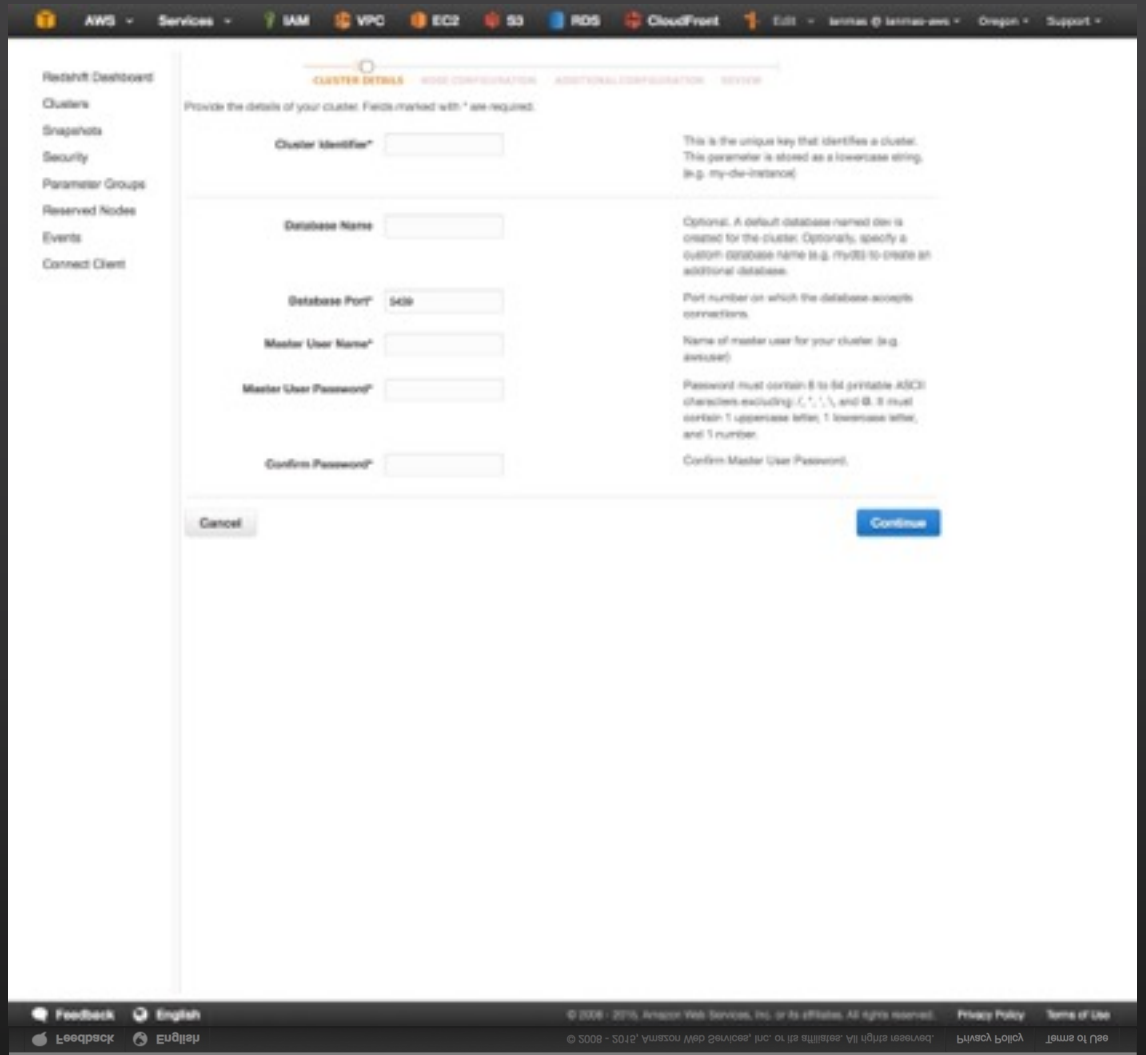
Creating an Amazon Redshift Data Warehouse via the AWS Console



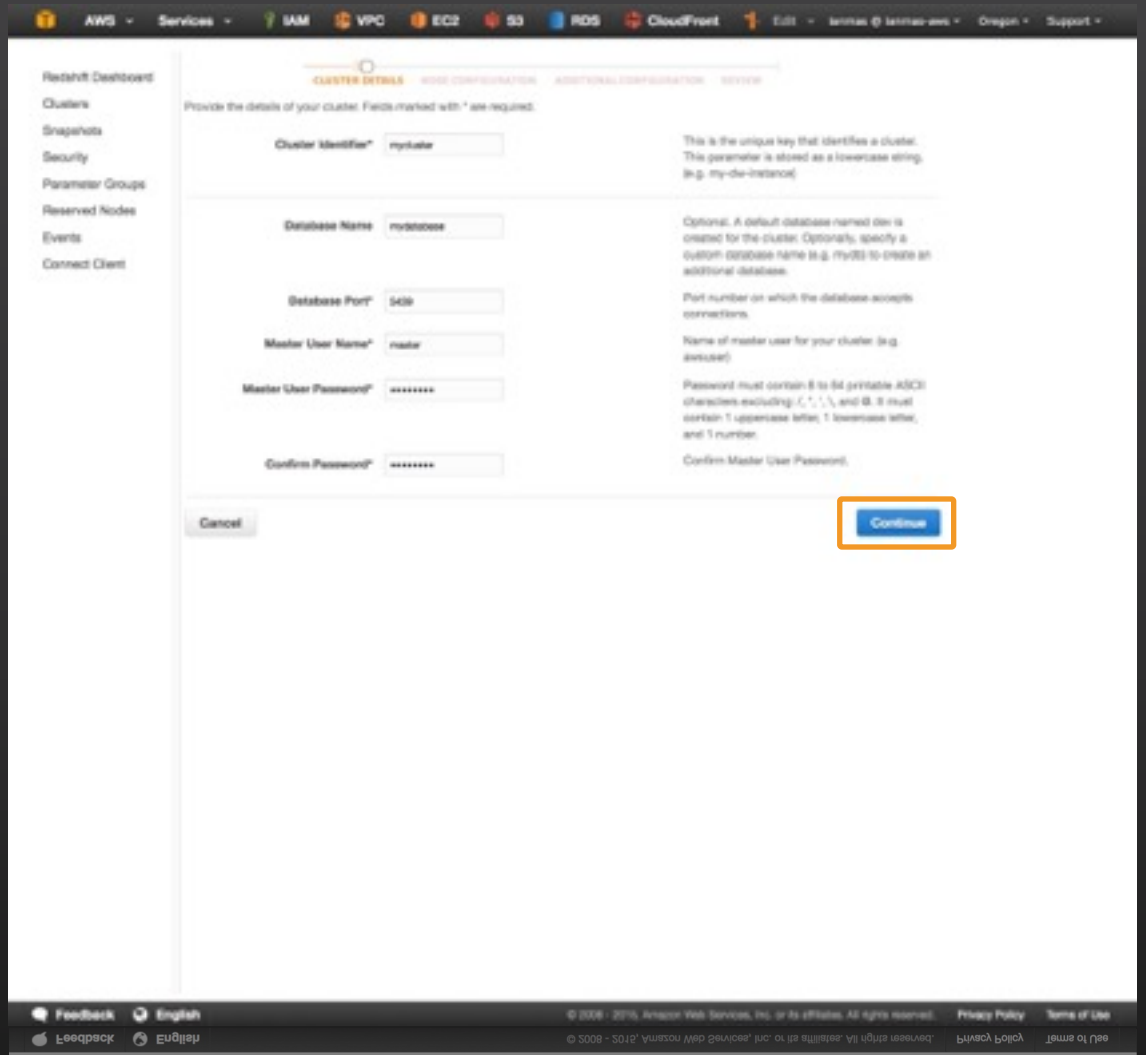
Creating an Amazon Redshift Data Warehouse via the AWS Console



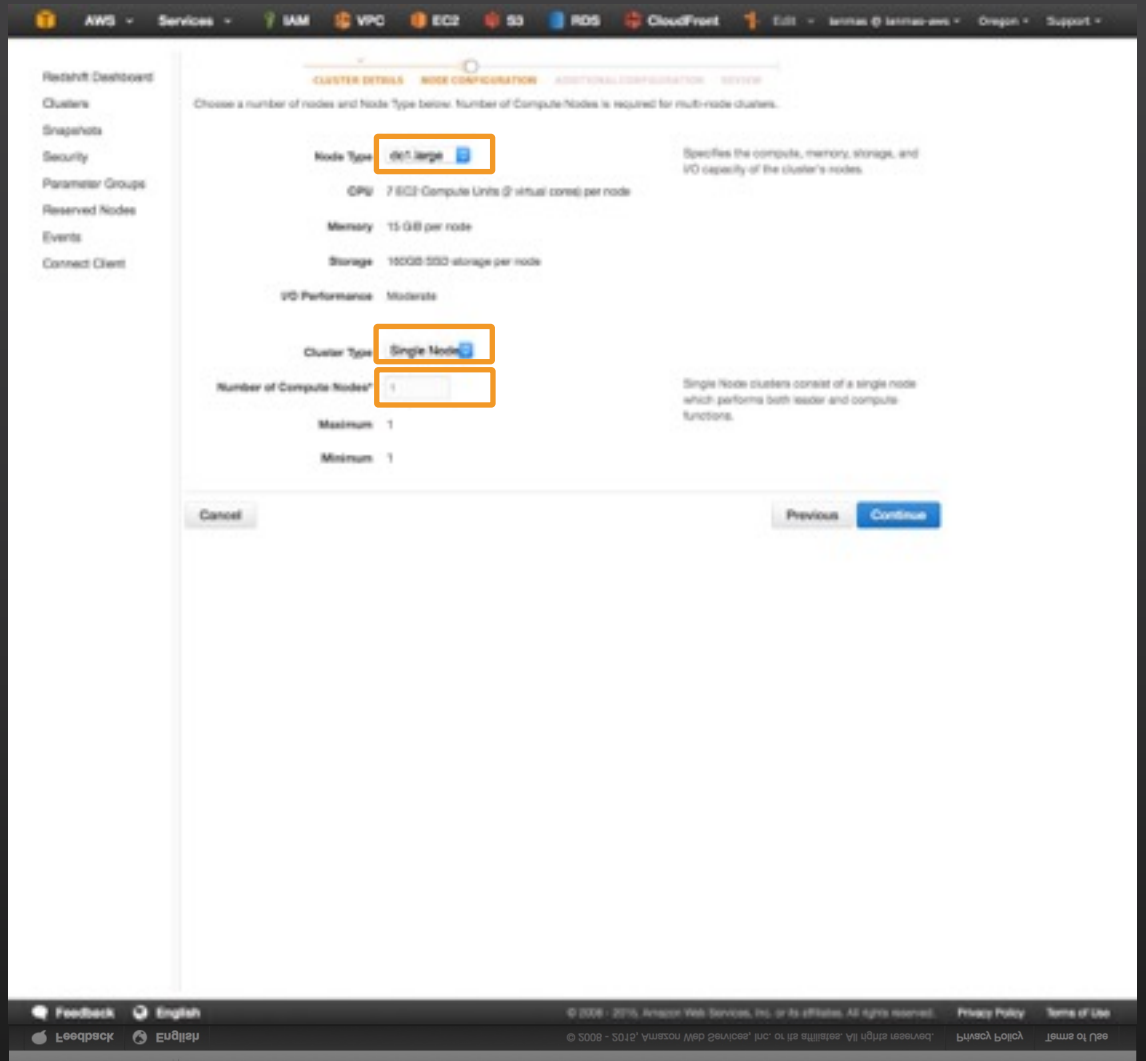
Creating an Amazon Redshift Data Warehouse via the AWS Console



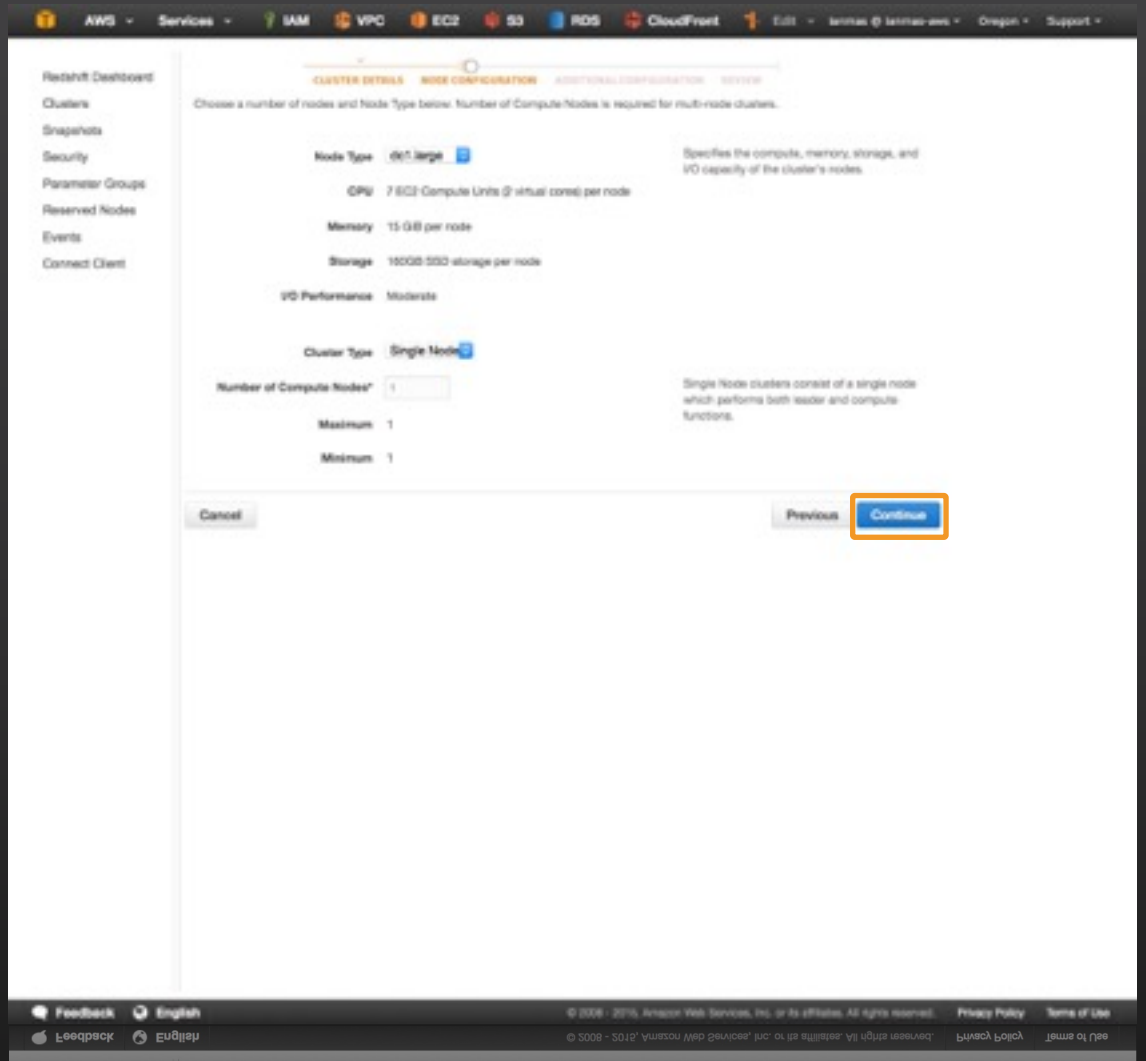
Creating an Amazon Redshift Data Warehouse via the AWS Console



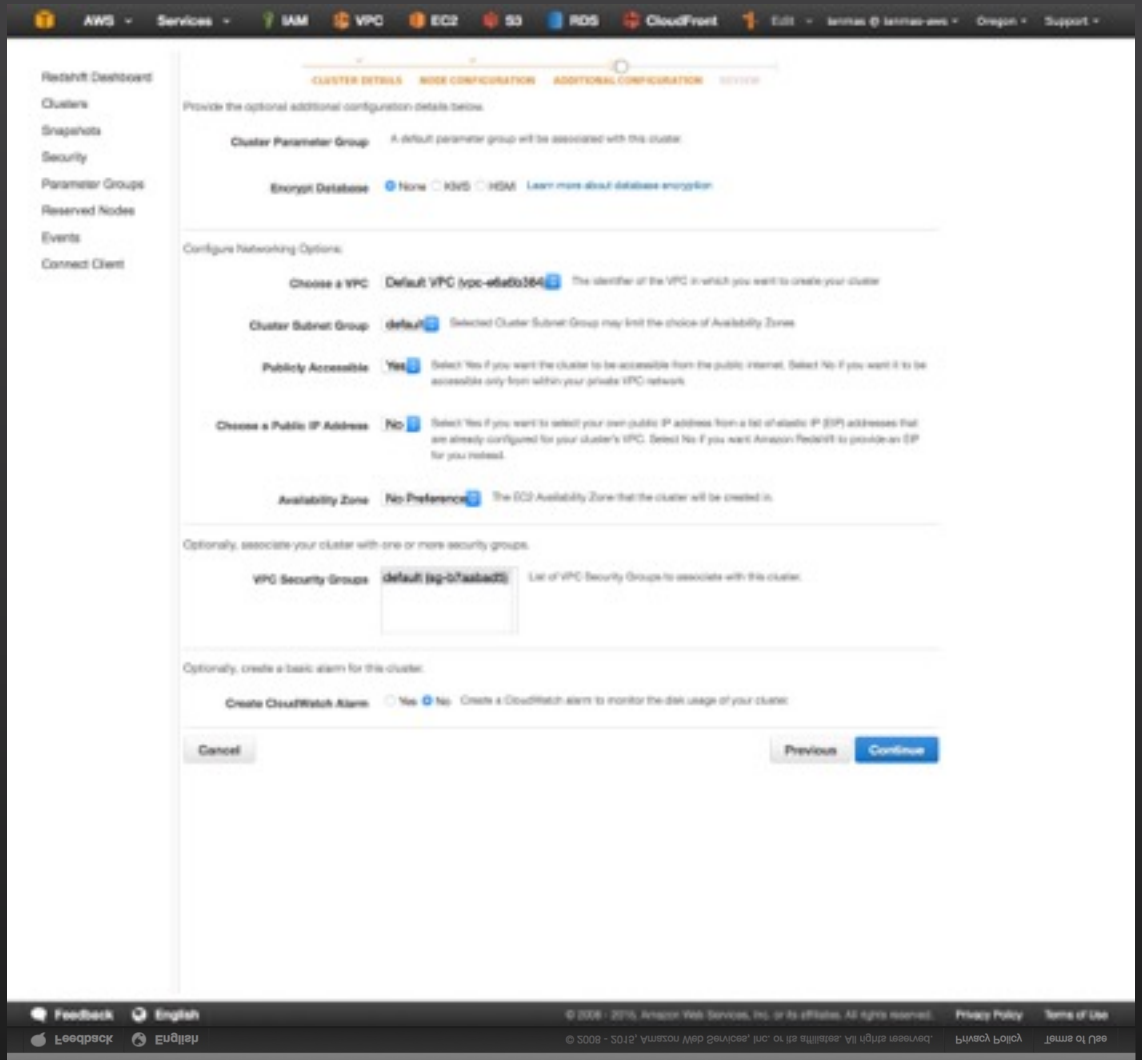
Creating an Amazon Redshift Data Warehouse via the AWS Console



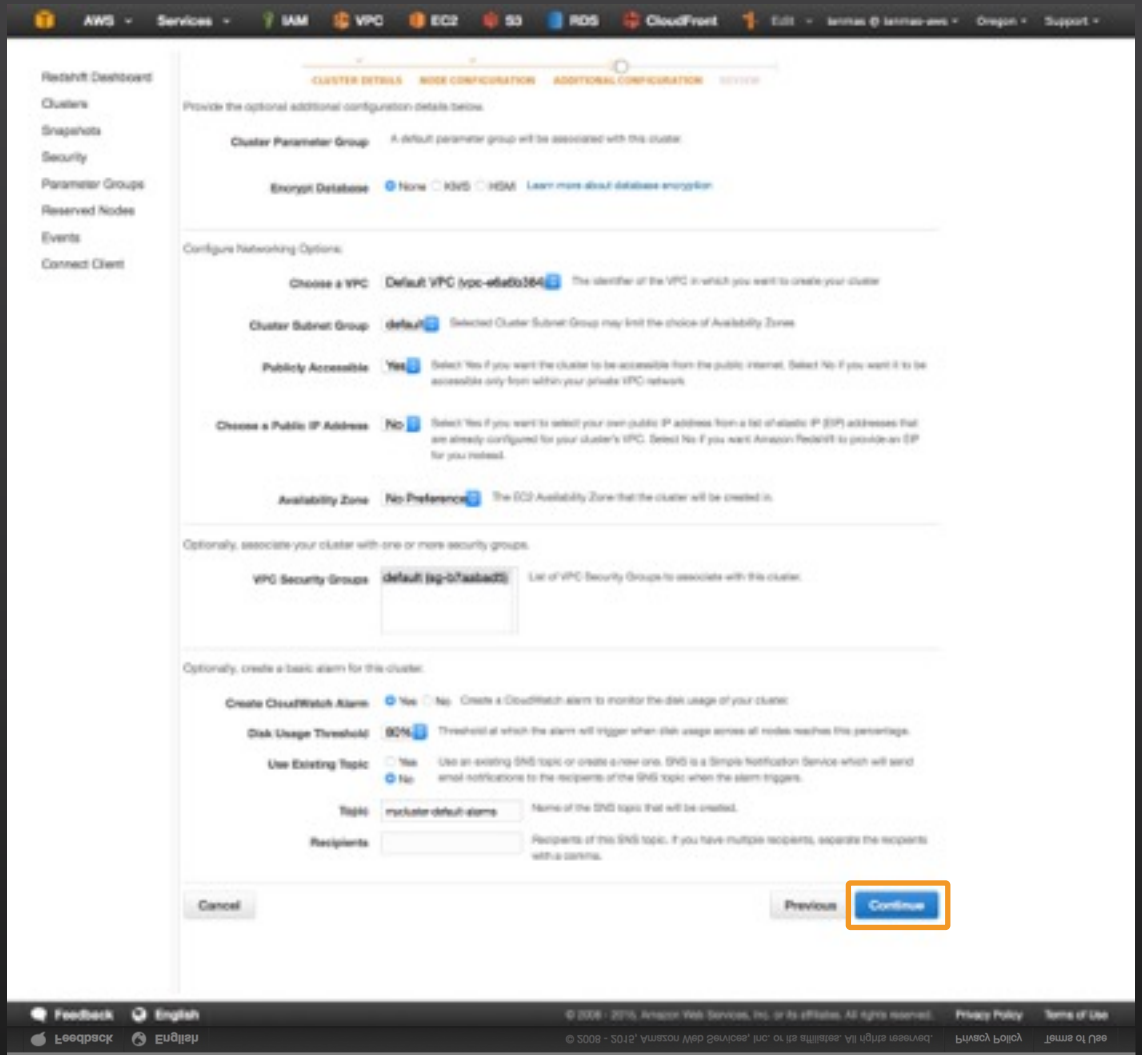
Creating an Amazon Redshift Data Warehouse via the AWS Console



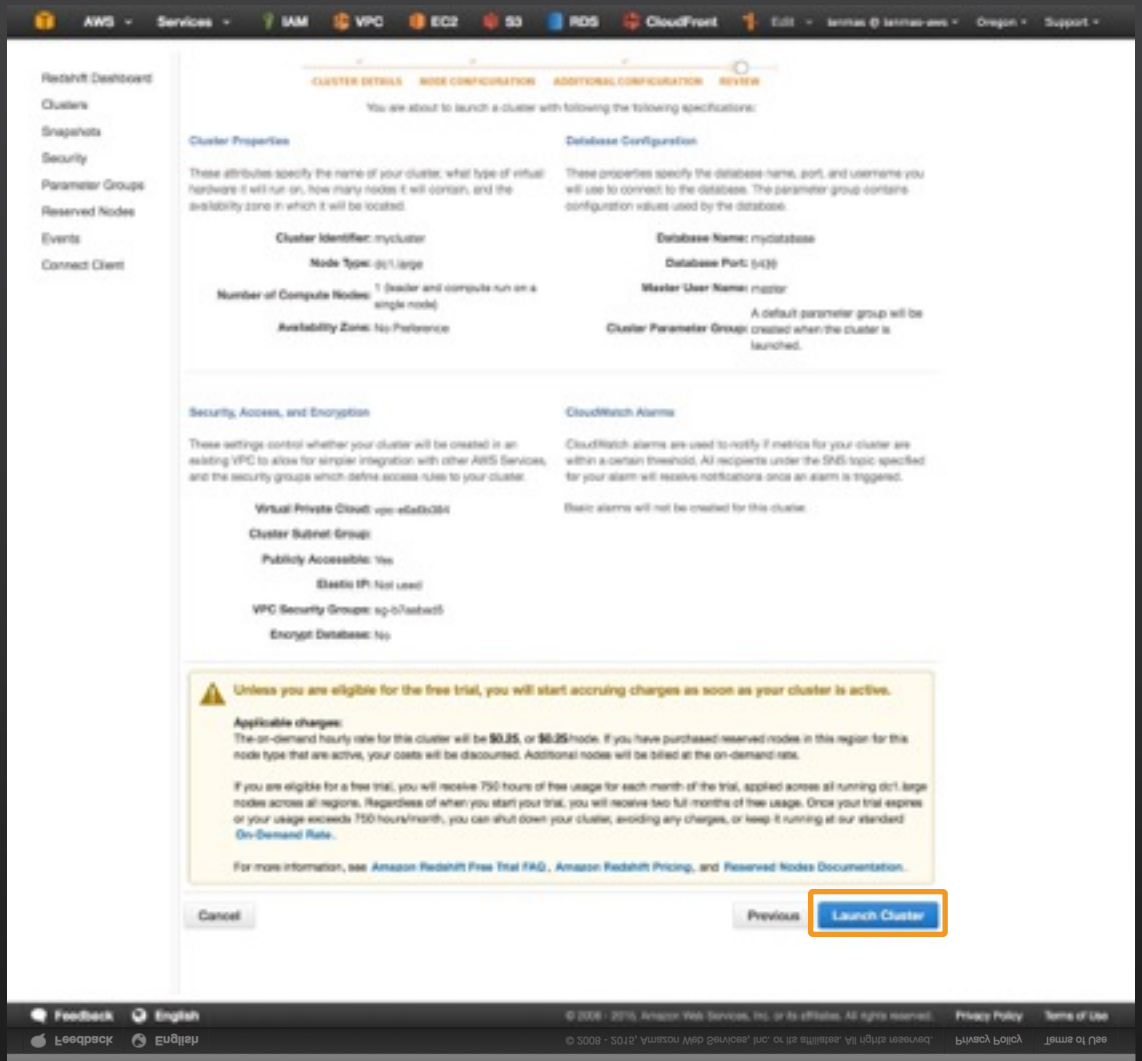
Creating an Amazon Redshift Data Warehouse via the AWS Console



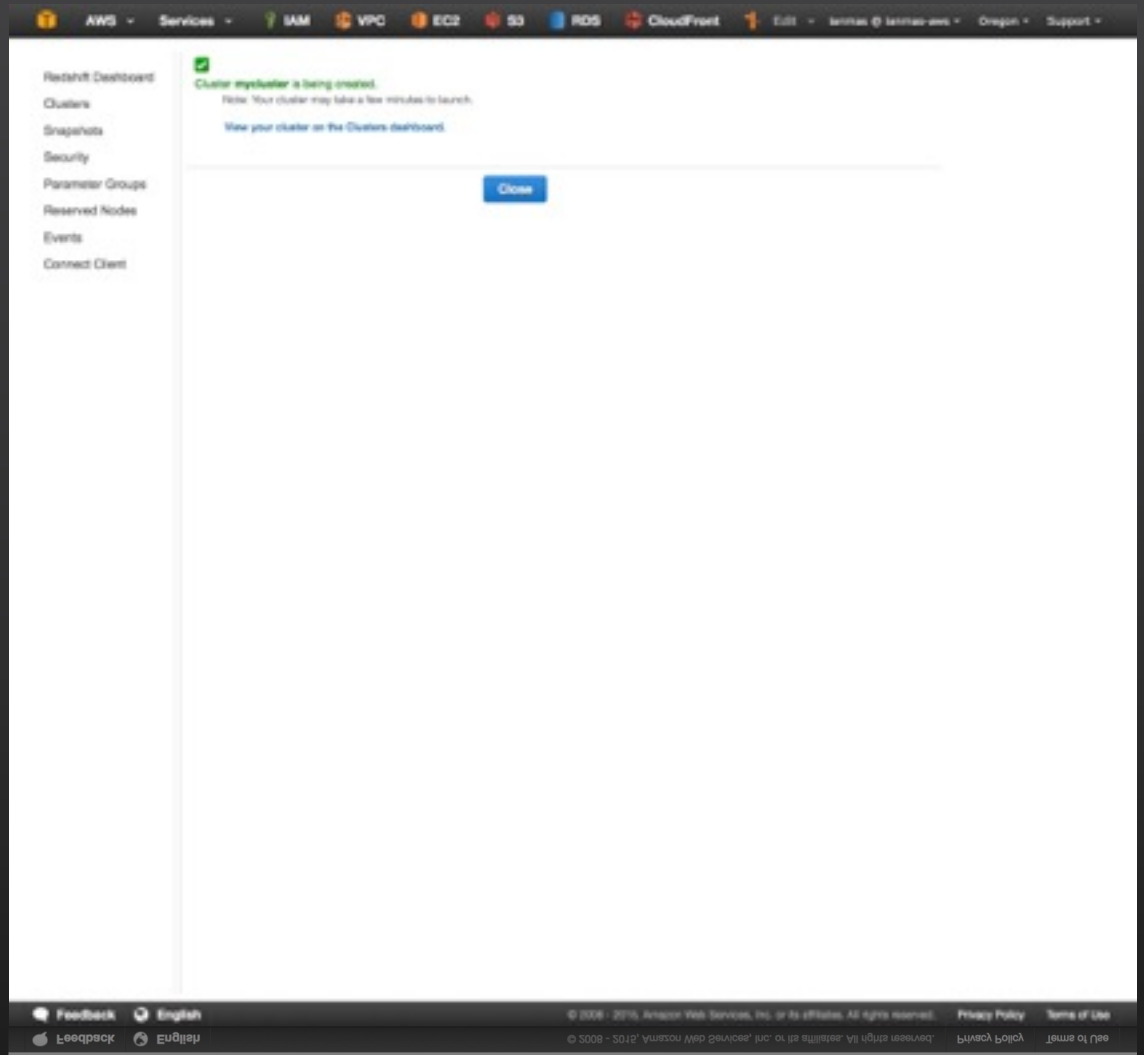
Creating an Amazon Redshift Data Warehouse via the AWS Console



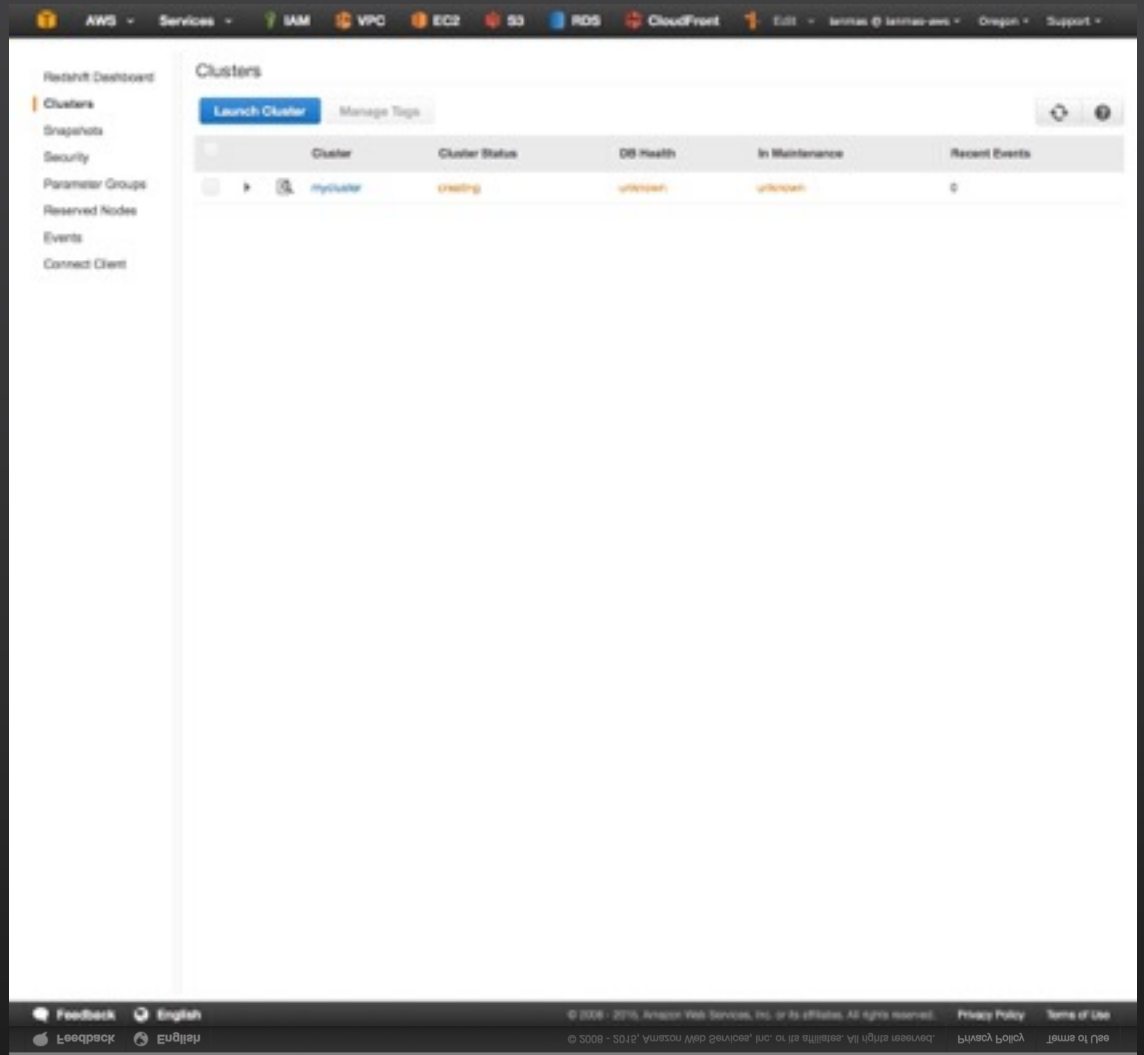
Creating an Amazon Redshift Data Warehouse via the AWS Console



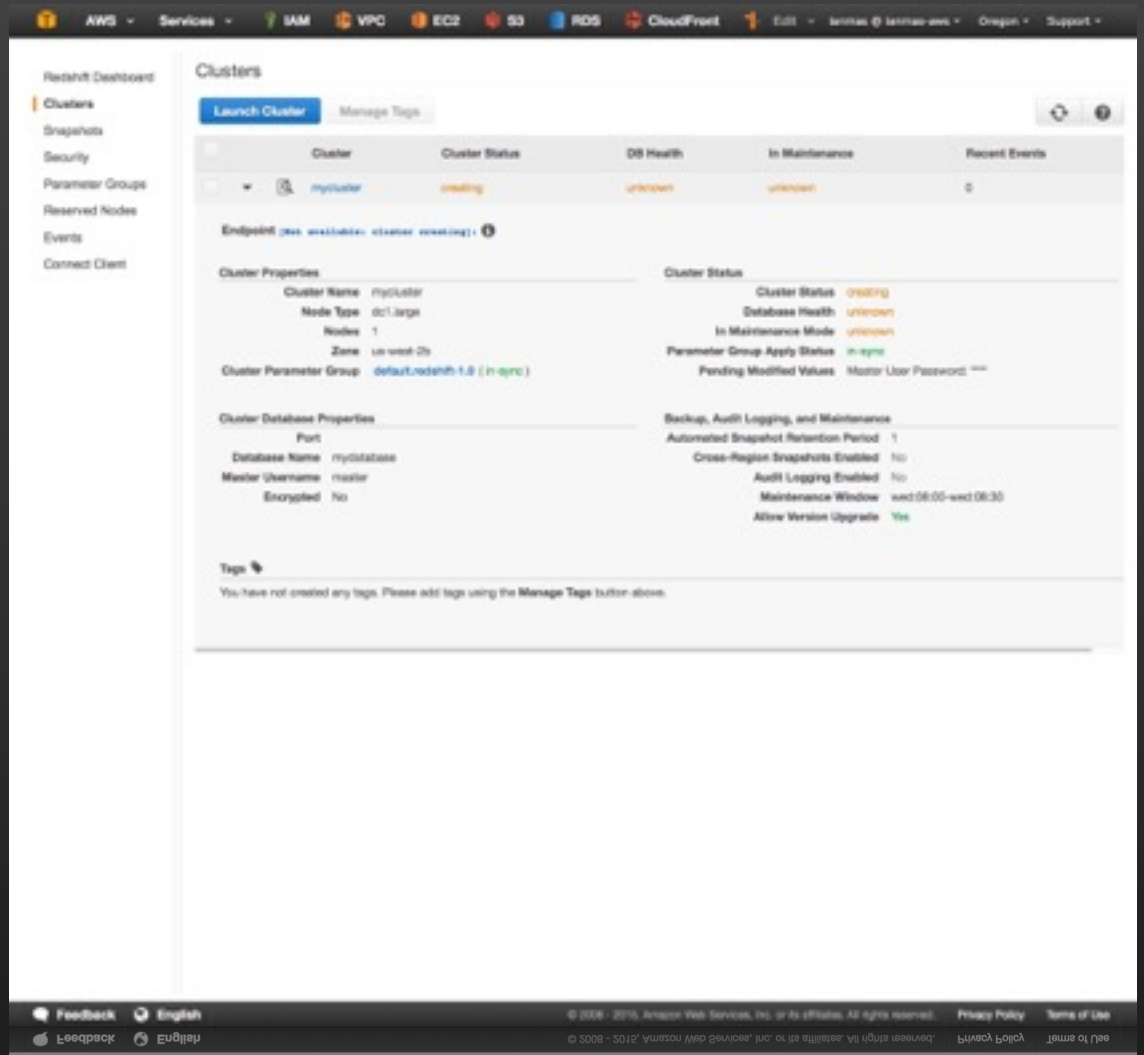
Creating an Amazon Redshift Data Warehouse via the AWS Console



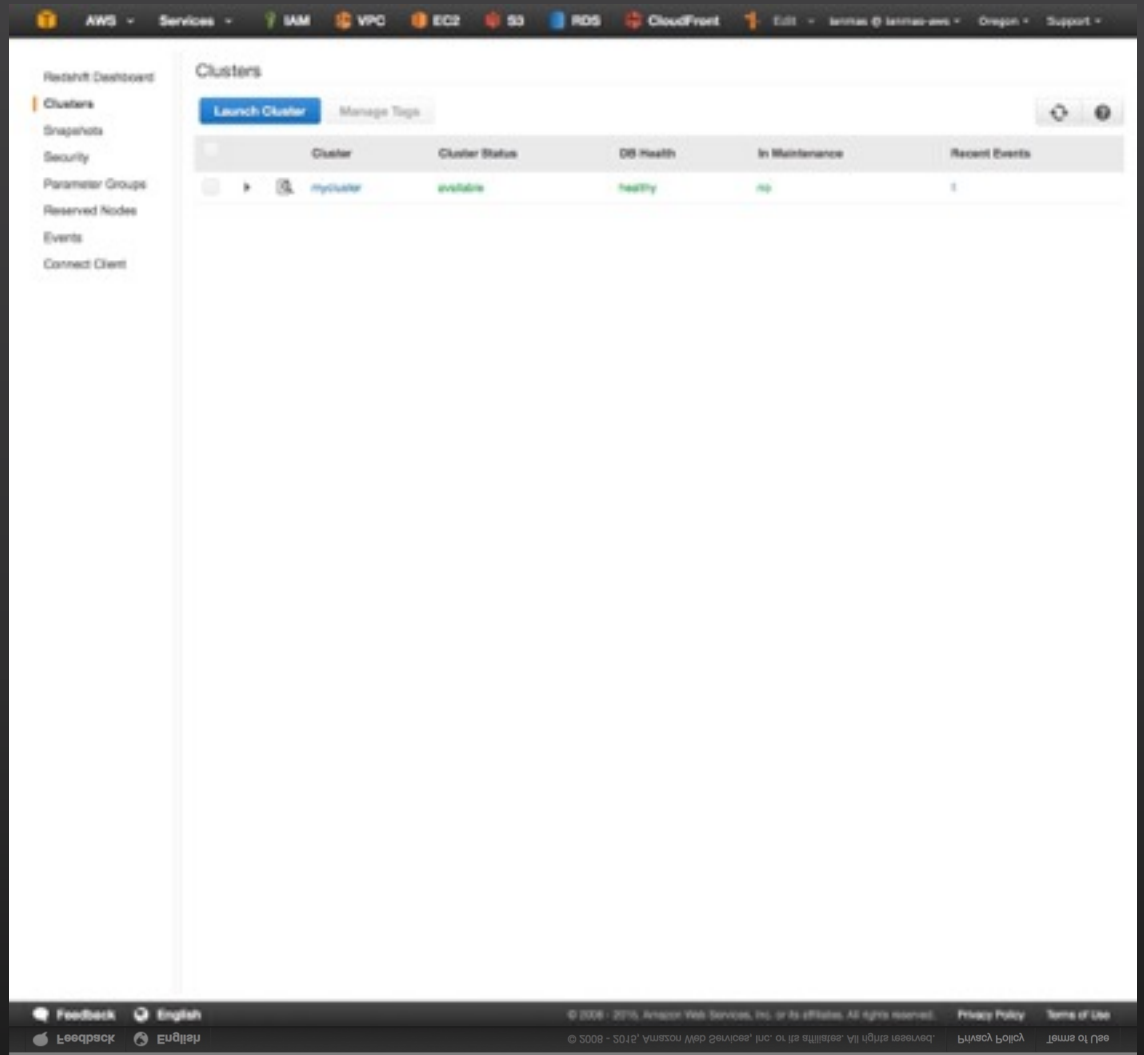
Creating an Amazon Redshift Data Warehouse via the AWS Console



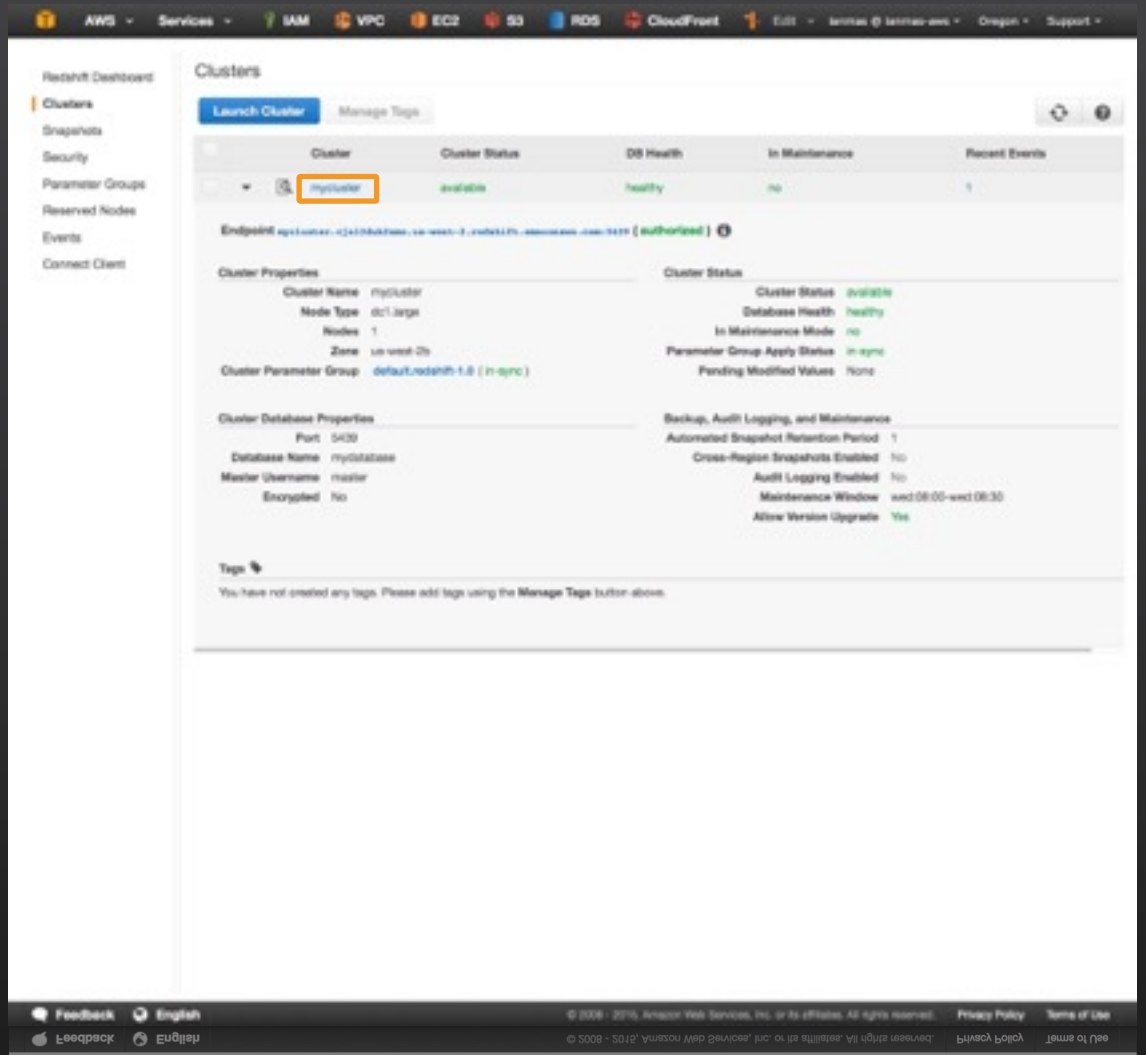
Creating an Amazon Redshift Data Warehouse via the AWS Console



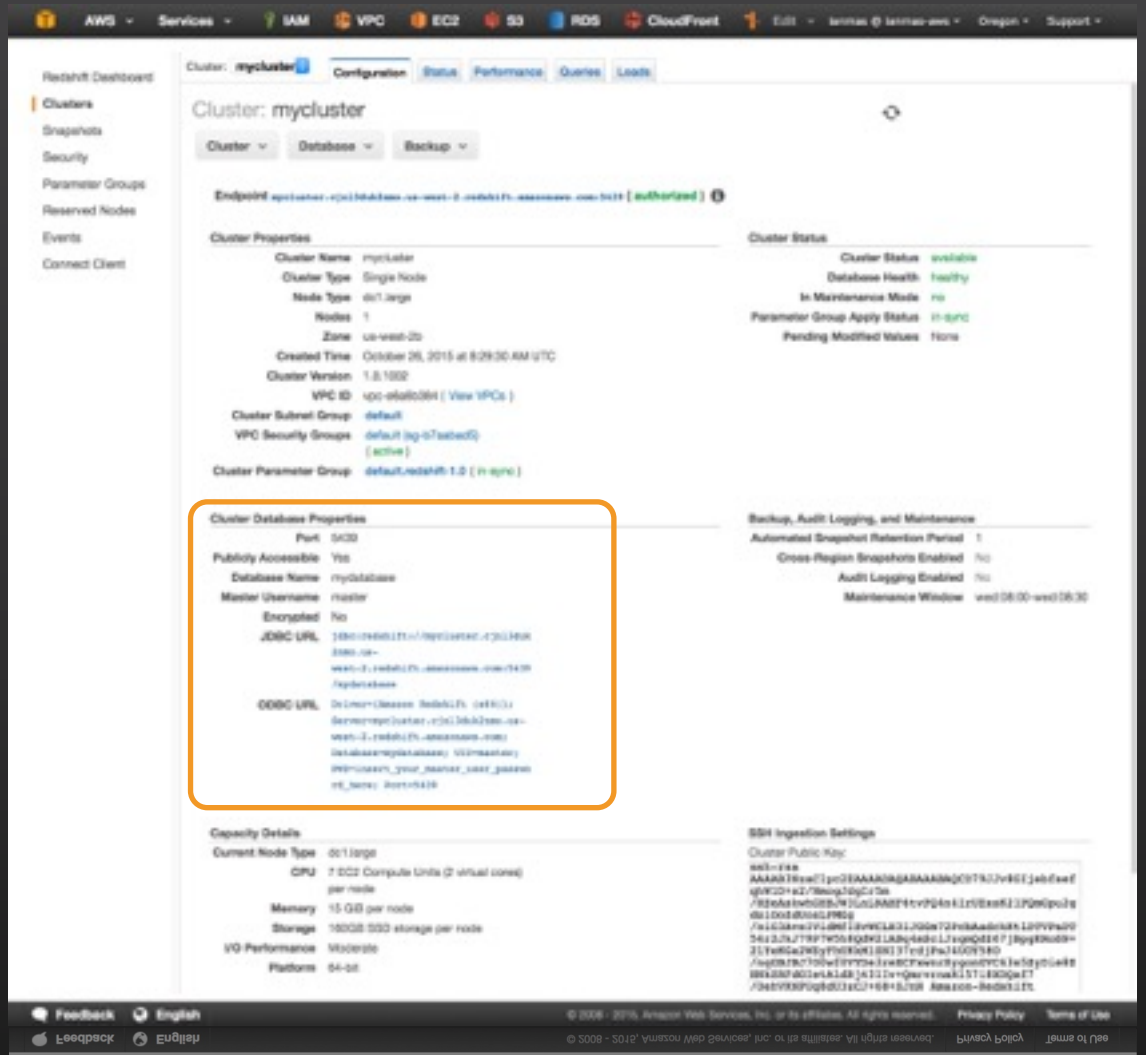
Creating an Amazon Redshift Data Warehouse via the AWS Console



Creating an Amazon Redshift Data Warehouse via the AWS Console



Creating an Amazon Redshift Data Warehouse via the AWS Console



Getting Started with Amazon Redshift

- 2 Month free trial
- Getting Started tutorial
- Amazon Redshift System Overview
- Guides for table design, loading data & query design
- Connect using industry standard OBDC/JDBC connections
- Many BI & ETL vendors have certified Amazon Redshift

http://aws.amazon.com/redshift/getting-started/

TABLE DESIGN

Choosing the optimal table design

- Compression Encodings
- Choosing the right data types
- Distributing and sorting data

Choosing a Column Compression Type

- Compression is a key tool improving performance in modern data warehouse systems to reduce the size of data that is read from storage, minimise IO & improve query performance
- The COPY command used to load data in Amazon Redshift will perform compression analysis
- We strongly recommend using the COPY command to apply automatic compression

http://docs.aws.amazon.com/redshift/latest/dg/t_Compressing_data_on_disk.html

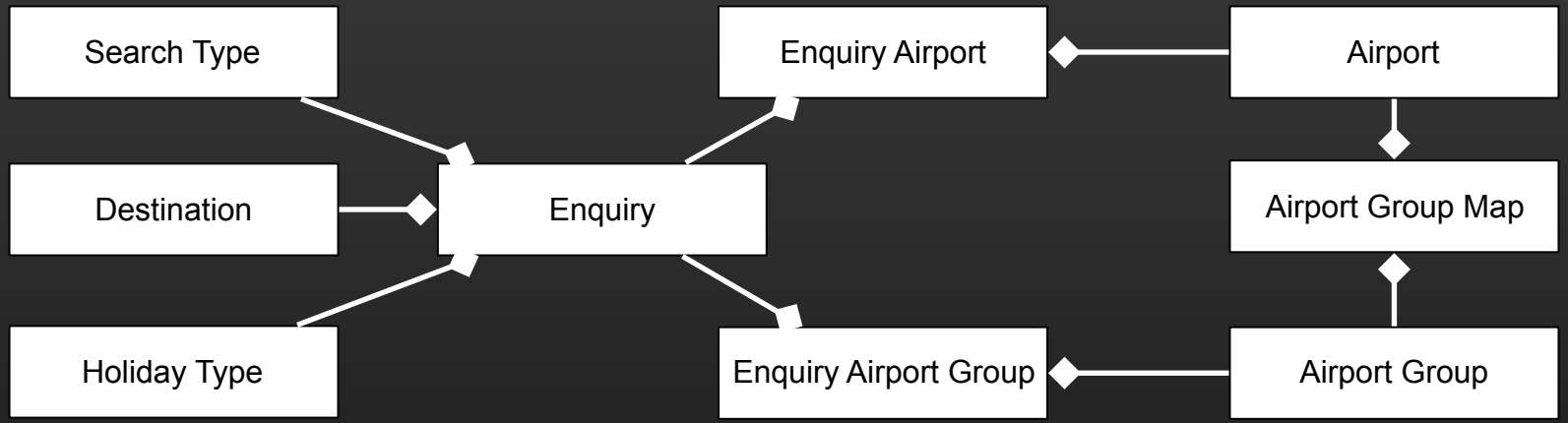
Data Types

Data Type	Aliases	Description
SMALLINT	INT2	Signed two-byte integer
INTEGER	INT, INT4	Signed four-byte integer
BIGINT	INT8	Signed eight-byte integer
DECIMAL	NUMERIC	Exact numeric of selectable precision
REAL	FLOAT4	Single precision floating-point number
DOUBLE PRECISION	FLOAT8, FLOAT	Double precision floating-point number
BOOLEAN	BOOL	Logical Boolean (true/false)
CHAR	CHARACTER, NCHAR, BPCHAR	Fixed-length character string
VARCHAR	CHARACTER VARYING, NVARCHAR, TEXT	Variable-length character string with a user-defined limit
DATE		Calendar date (year, month, day)
TIMESTAMP	TIMESTAMP WITHOUT TIME ZONE	Date and time (without time zone)

http://docs.aws.amazon.com/redshift/latest/dg/c_Supported_data_types.html

Your Schema

- Expressed in 3rd Normal Form



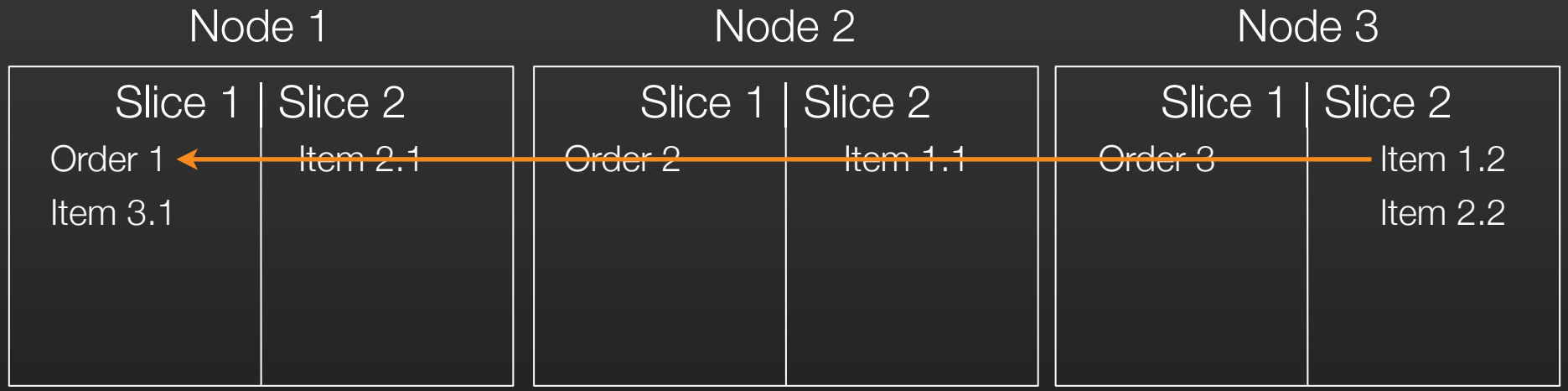
Distributing Data

- Redshift is a distributed system:
  - A cluster contains a leader node & compute nodes
  - A compute node contains slices (one per core)
  - A slice contains data
- Slices are chosen based on two types of distribution:
  - Round Robin (automated)
  - Based on a distribution key (hash of a defined column)
- Queries run on all slices in parallel: optimal query throughput can be achieved when data is evenly spread across slices

Unoptimised Distribution

Order 1: Dave Smith, Total £195

Item 1.1: Order 1, Kindle Fire HD 7”, £159
Item 1.2: Order 1, Kindle Fire Case, £36

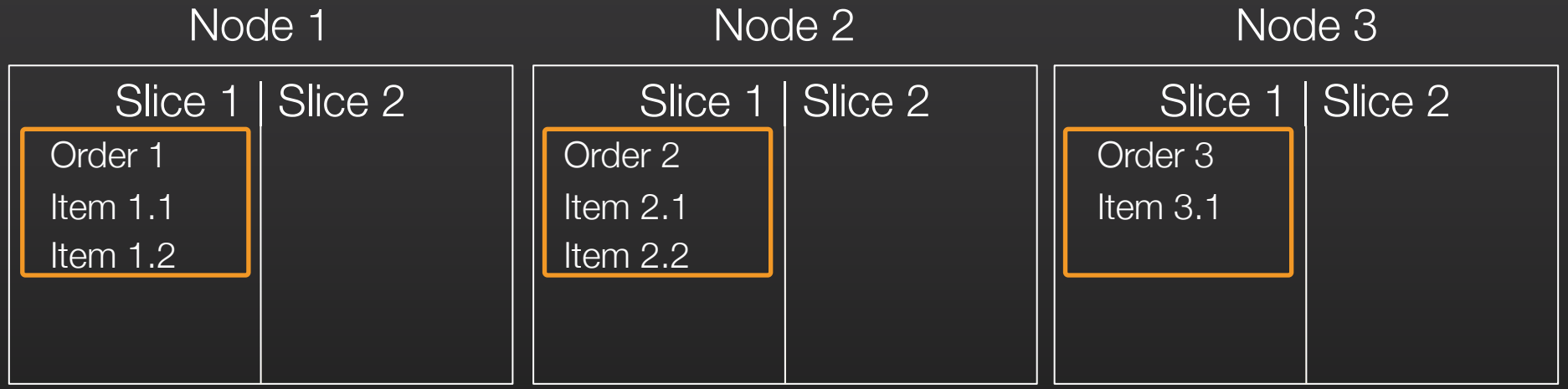


Default (No Distribution Key, Round Robin Order)

Optimised Distribution

Order 1: Dave Smith, Total £195

Item 1.1: Order 1, Kindle Fire HD 7”, £159
Item 1.2: Order 1, Kindle Fire Case, £36



Customised (ORDERS.ORDER_ID DISTKEY, ITEMS.ORDER_ID DISTKEY)

Choosing a Distribution Key

- Frequently Joined
  - By most commonly run queries
  - By queries which consume the most CPU
- High Cardinality
  - Large number of discrete values
- Low Skew
  - Uniform Distribution
  - No Hotspots
  - Query STV_BLOCKLIST for Skew Factor

Sorting Data

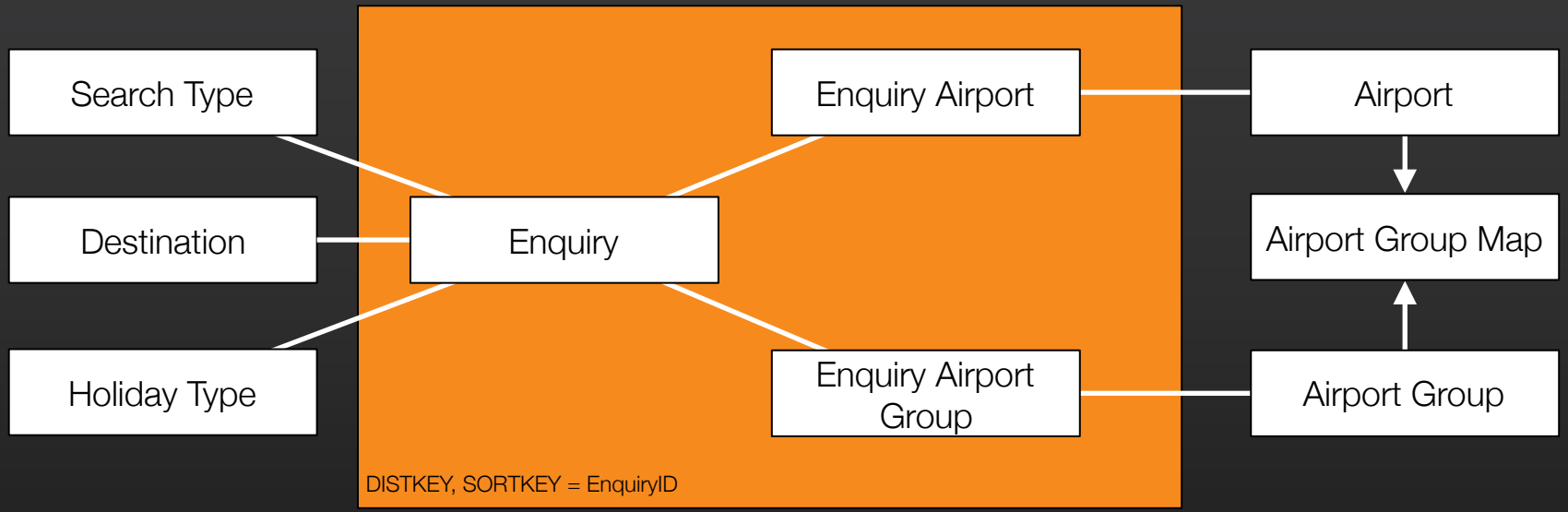
- In the slices (on disk), the data is sorted by a sort key (if none uses the insertion order)
- Choose a sort key that is frequently used in your queries
  - As a query predicate (date, identifier, …)
  - As a join parameter (it can also be the hash key)
- The sort key allows Redshift to avoid reading entire blocks based on predicates
- E.g.: a table containing a timestamp sort key, and where only recent data is accessed, will skip blocks containing “old” data

Schema Design

- Optimizing a database for querying
  - Analyse using Automatic Compression to ensure optimal IO
  - Co-locate frequently joined tables: use distribution key wisely (avoids data transfers between nodes)
  - For joined tables, use sort keys on the joined columns, allowing fast merge joins
  - Compression allows you to de-normalize without penalizing storage, simplifying queries and limiting joins

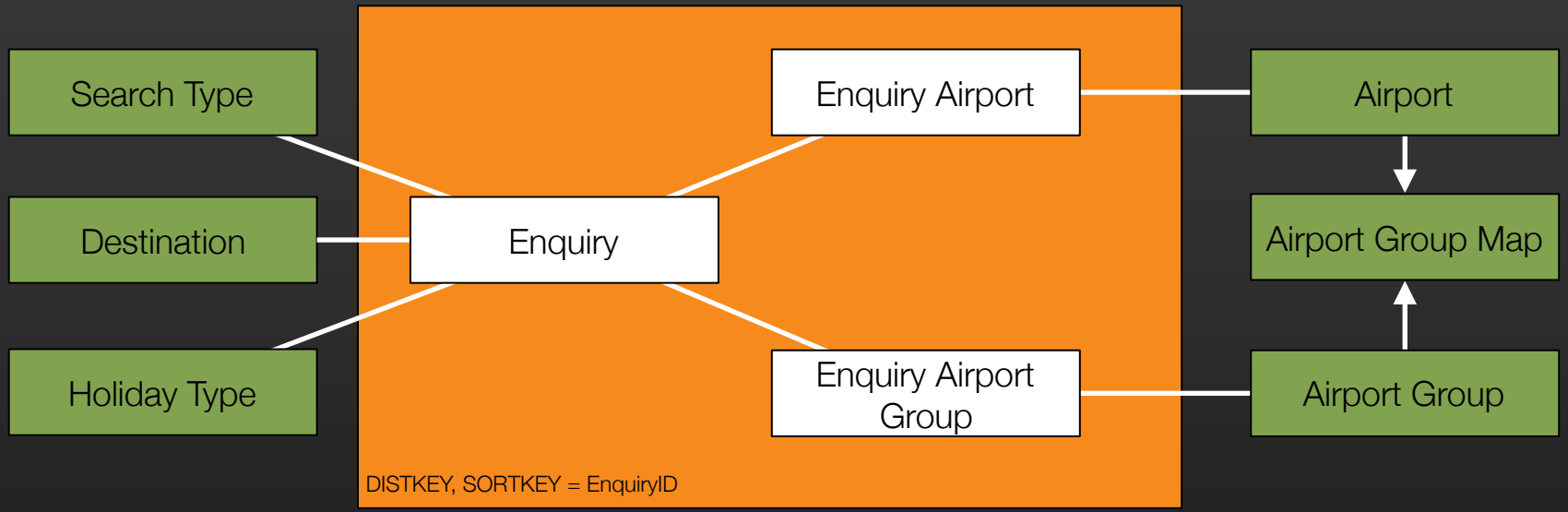
Possible Schema Optimisations

- Denormalise commonly used join attributes onto large tables



Possible Schema Optimisations

- Cache Small Tables with DISTSTYLE ALL



http://docs.aws.amazon.com/redshift/latest/dg/r_CREATE_TABLE_NEW.html

Vacuuming Tables

- Clean up tables after a bulk delete, a load, or a series of incremental updates with the VACUUM command, either against the entire database or against individual tables

http://docs.aws.amazon.com/redshift/latest/dg/t_Reclaiming_storage_space202.html

Learning more about Table Design

- Table Design Documentation

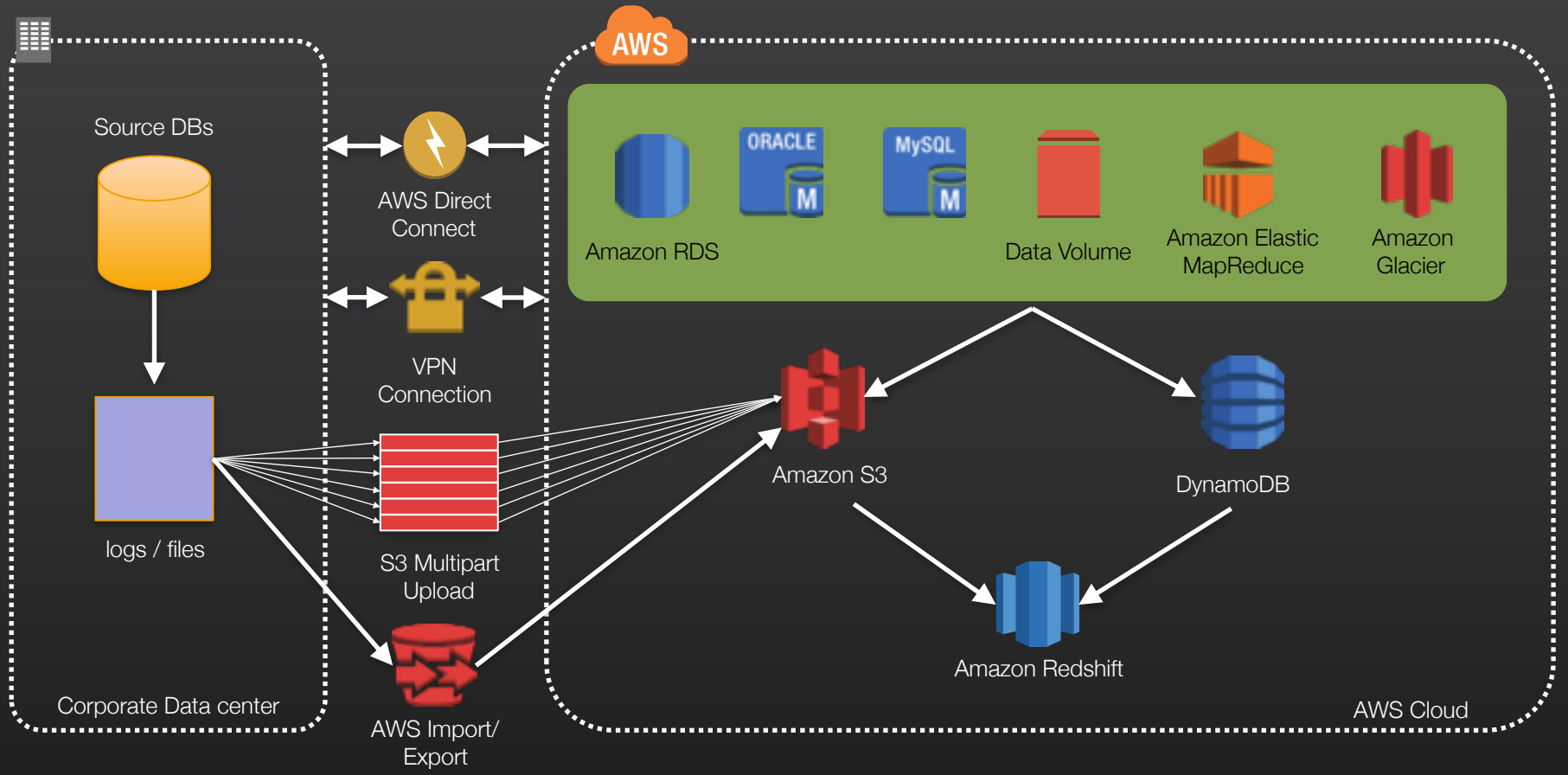
  http://docs.aws.amazon.com/redshift/latest/dg/t_Creating_tables.html

- Table Design Tutorial

  http://docs.aws.amazon.com/redshift/latest/dg/tutorial-tuning-tables.html

DATA LOADING

Amazon Redshift Loading Data Overview

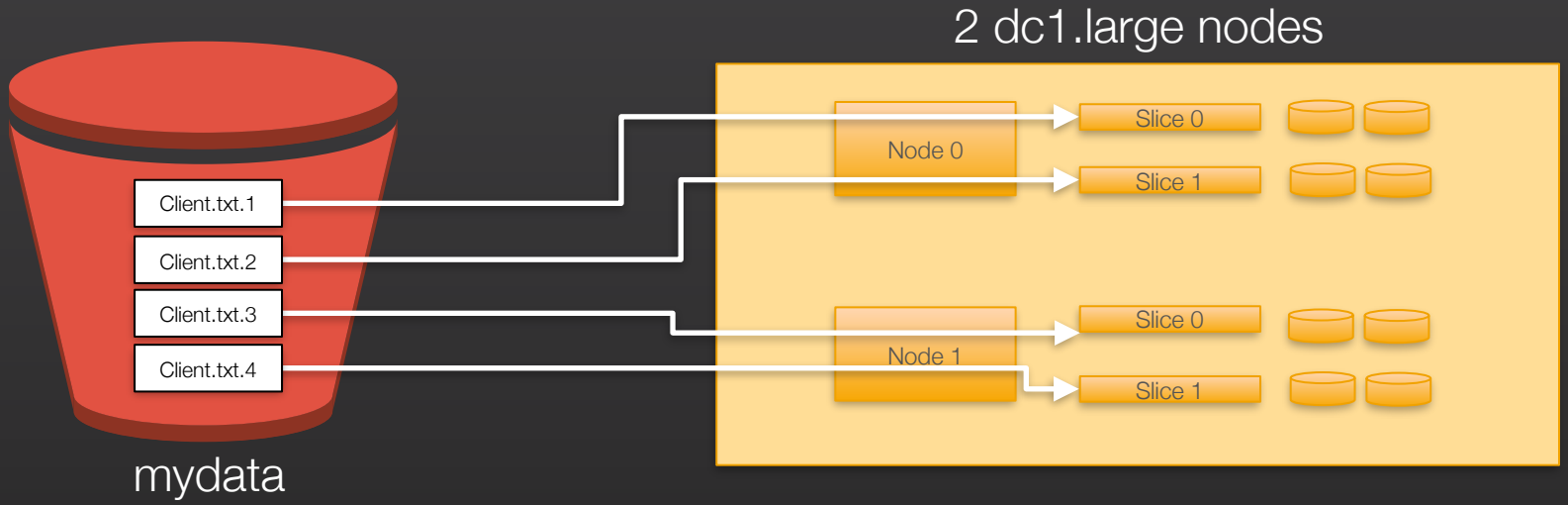


Sharing Your Data

- Data Dumped from DB
- CSV Files Encrypted
  - Review Encryption Code
- Uploaded to S3
  - Bucket <bucket>
- Data shared with AWS

```
{
  "Id": "Policy1234567890",
  "Statement": [
    {
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Effect": "Allow",
      "Principal": {
        "AWS": ["arn:aws:iam::887210671223:root"]
      },
      "Resource": [
        "arn:aws:s3:::<bucket-name>",
        "arn:aws:s3:::<bucket-name>/*"
      ],
      "Sid": "Stmt0987654321"
    }
  ]
}
```

Splitting Data Files




```
copy customer
from ‘s3://mydata/client.txt’
credentials ‘aws_access_key_id=<your-access-key>; aws_secret_access_key=<your_secret_key>’
delimiter ‘|’;
```

http://docs.aws.amazon.com/redshift/latest/dg/t_Loading_data.html

Typical Load Issues

- Mismatch between data types in table and values in input data fields
- Mismatch between number of columns in table and number of fields in input data
- Mismatched quotes
  - Redshift supports both single and double quotes; however, these quotes must be balanced appropriately
- Incorrect format for date/time data in input files
  - Use DATEFORMAT and TIMEFORMAT to control
- Out-of-range values in input files (for numeric columns)
- Number of distinct values for a column exceeds the limitation for its compression encoding

Direct SQL

- Redshift supports standard DML commands
  - INSERT, UPDATE, DELETE
- Redshift does not support single-command merge (upsert) statement
  - Load data into a staging table
  - Joining the staging table with the target table
    - UPDATE data where row exists
    - INSERT where no row exists
- All Direct SQL Commands go via Leader Node

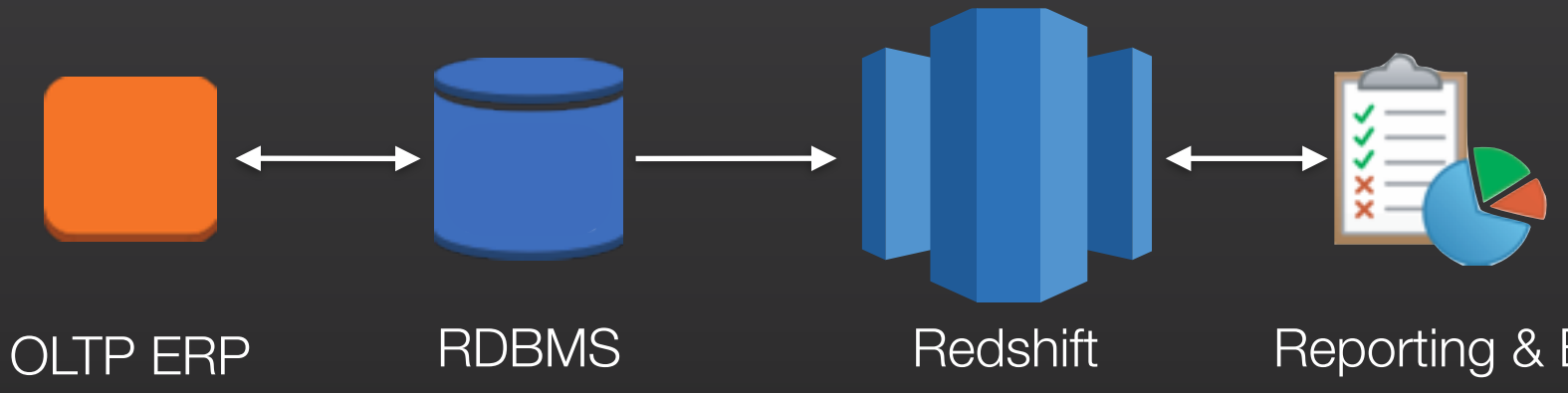
Data Loading Best Practices

- Use a COPY Command to load data
- Use a single COPY command
- Split your data into multiple files
- Compress your data files with GZIP
- Use multi-row inserts if COPY is not possible
- Bulk insert operations (INSERT INTO…SELECT and CREATE TABLE AS) provide high performance data insertion

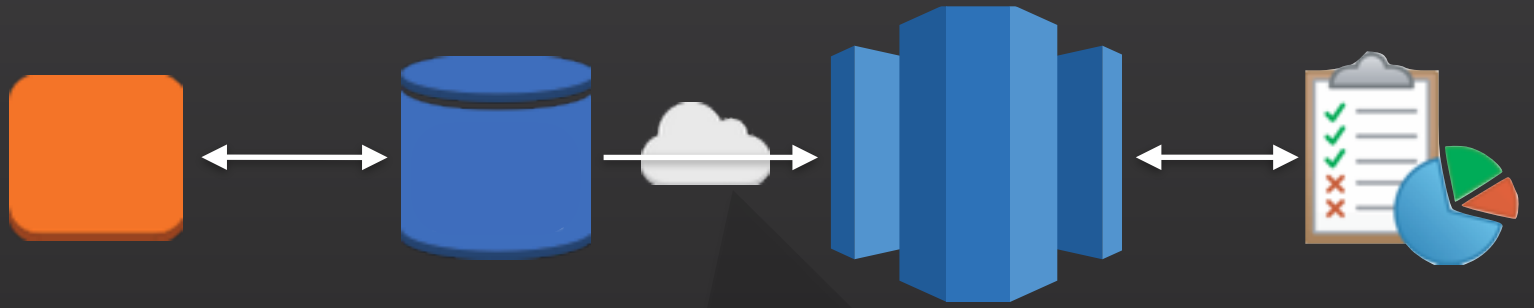
Data Loading Best Practices

- Load your data in sort key order to avoid needing to vacuum
- Organize your data as a sequence of time-series tables, where each table is identical but contains data for different time ranges
- Use staging tables to perform an upsert
- Run the VACUUM command whenever you add, delete, or modify a large number of rows, unless you load your data in sort key order
- Increase the memory available to a COPY or VACUUM by increasing wlm_query_slot_count
- Run the ANALYZE command whenever you've made a non-trivial number of changes to your data to ensure your table statistics are current

WORKING WITH DATA IN AMAZON REDSHIFT


OLTP ERP
RDBMS
Redshift
Reporting & BI

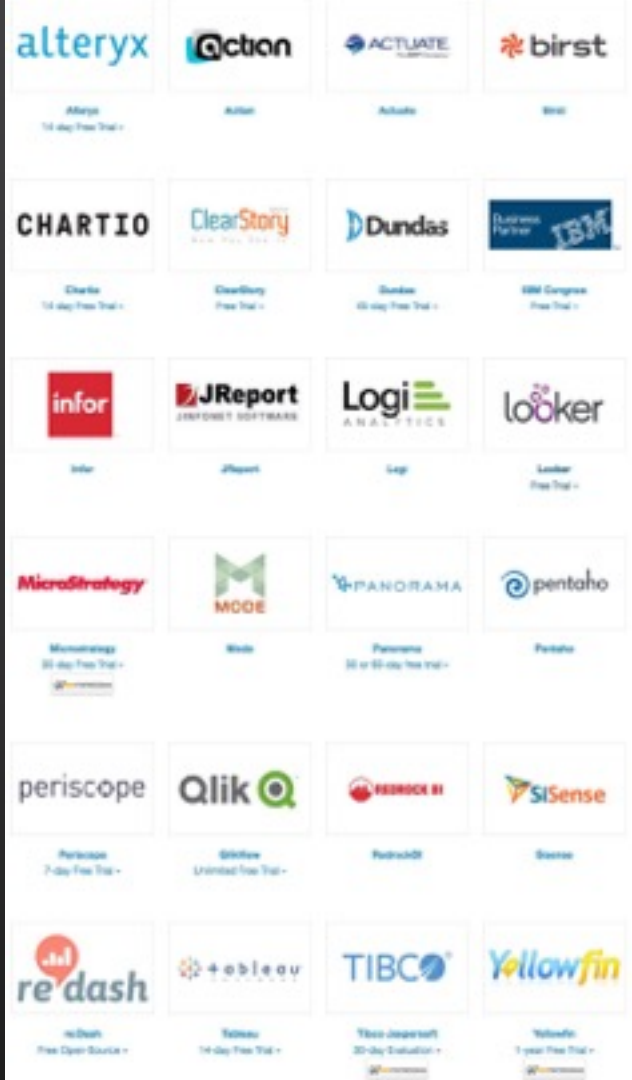
Data Integration



https://aws.amazon.com/redshift/partners/

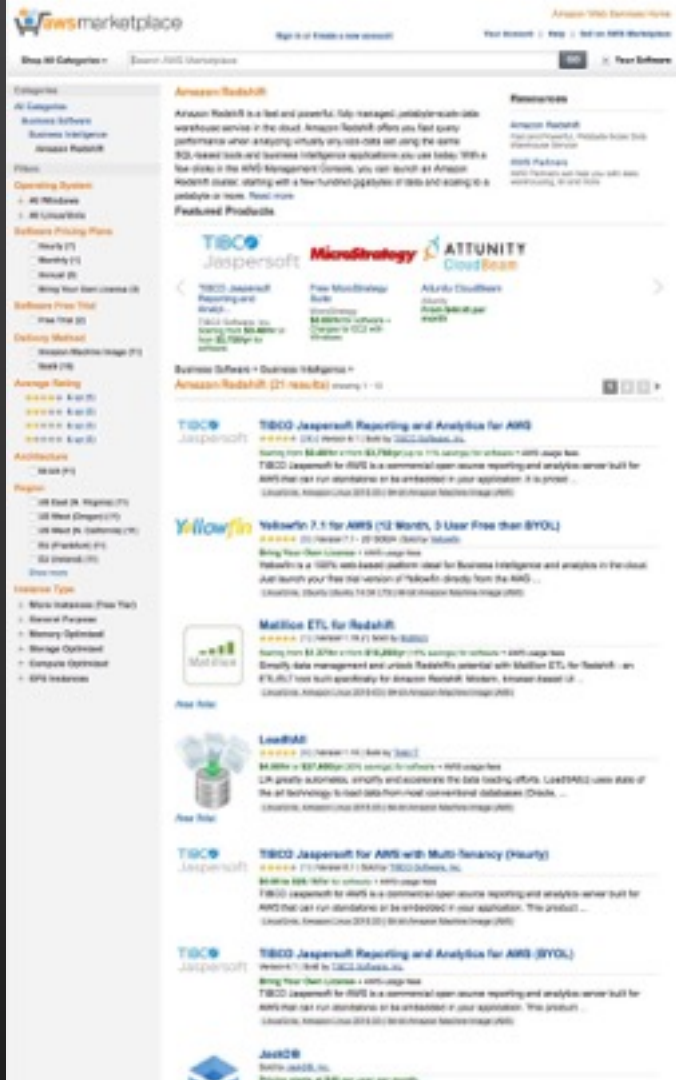
Business Intelligence & Visualisation

JDBC/ODBC



https://aws.amazon.com/redshift/partners/

https://aws.amazon.com/marketplace/redshift




Looker Analytics Platform - 10 Users, Multi-node Redshift (plus RDS)
Pricing Details
Product Description
Product Details
Resources
Usage Instructions
Support Details

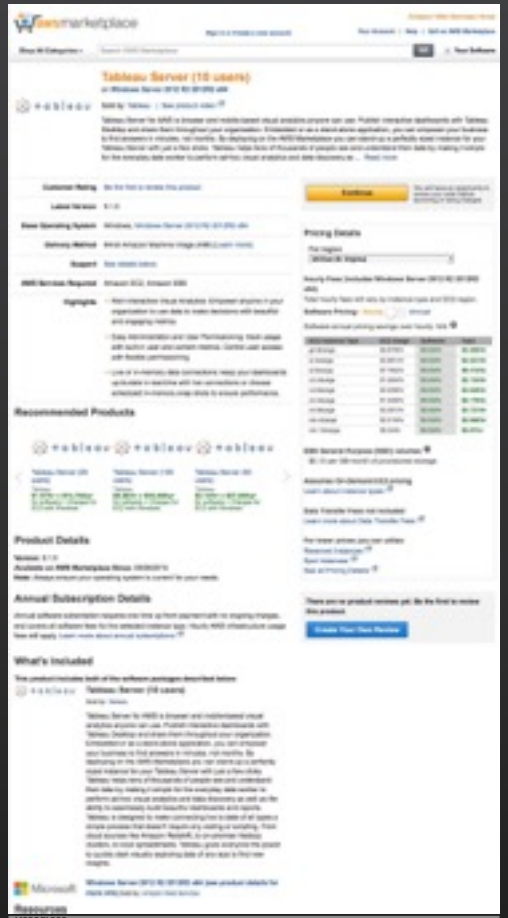


Tableau Server (10 users)
Pricing Details
Recommended Products
Product Details
Annual Subscription Details
What's Included


TIBCO Jaspersoft Reporting and Analytics for AWS
Pricing Details
Recommended Products
Product Description
Product Details
Annual Subscription Details
Resources
Usage Instructions

AMAZON QUICKSIGHT

New

FAST, EASY TO USE, CLOUD POWERED BUSINESS INTELLIGENCE

AVAILABLE IN PREVIEW NOW

BRINGING BUSINESS INTELLIGENCE TO ALL, WITH AMAZON QUICKSIGHT

FIRST ANALYSIS IN LESS THAN 60 SECONDS

BLAZING FAST QUERIES WITH A NEW IN-MEMORY QUERY ENGINE (SPICE)

DYNAMIC, BEAUTIFUL DATA VISUZALITIONS

SHARE LIVE AND SNAPSHOT ANALYSES WITH EVERYONE

INTEGRATE WITH DATA SOURCES ON AWS

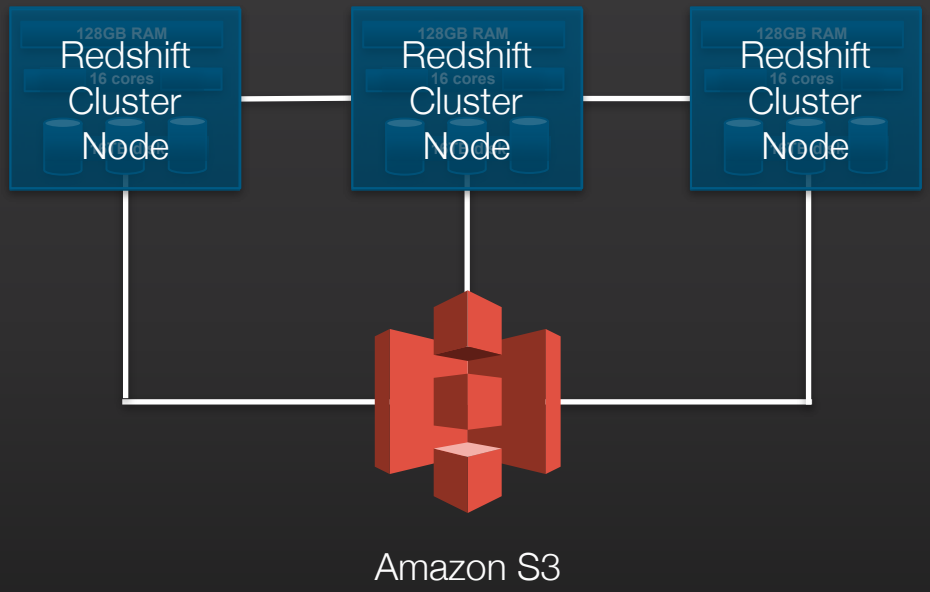
1/10TH THE COST OF OLD-GUARD BI TOOLS

Amazon QuickSight Launch & Demo

https://youtu.be/Tj0gW4XI6vU

BACKUP AND RESTORATION

Backup & Restoration

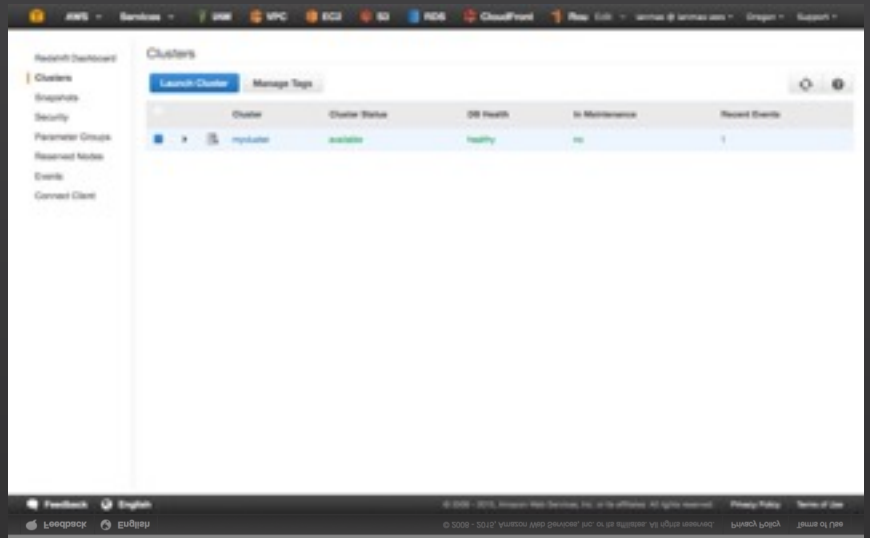


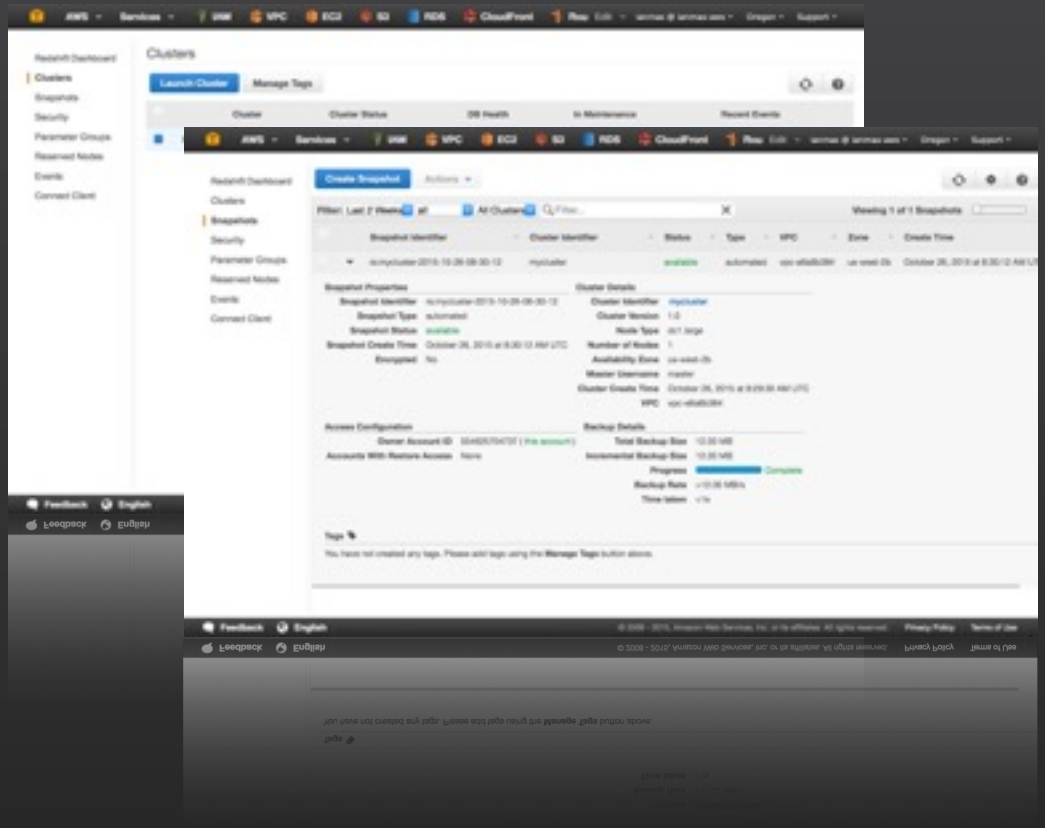
Backups to Amazon S3 are automatic, continuous & incremental

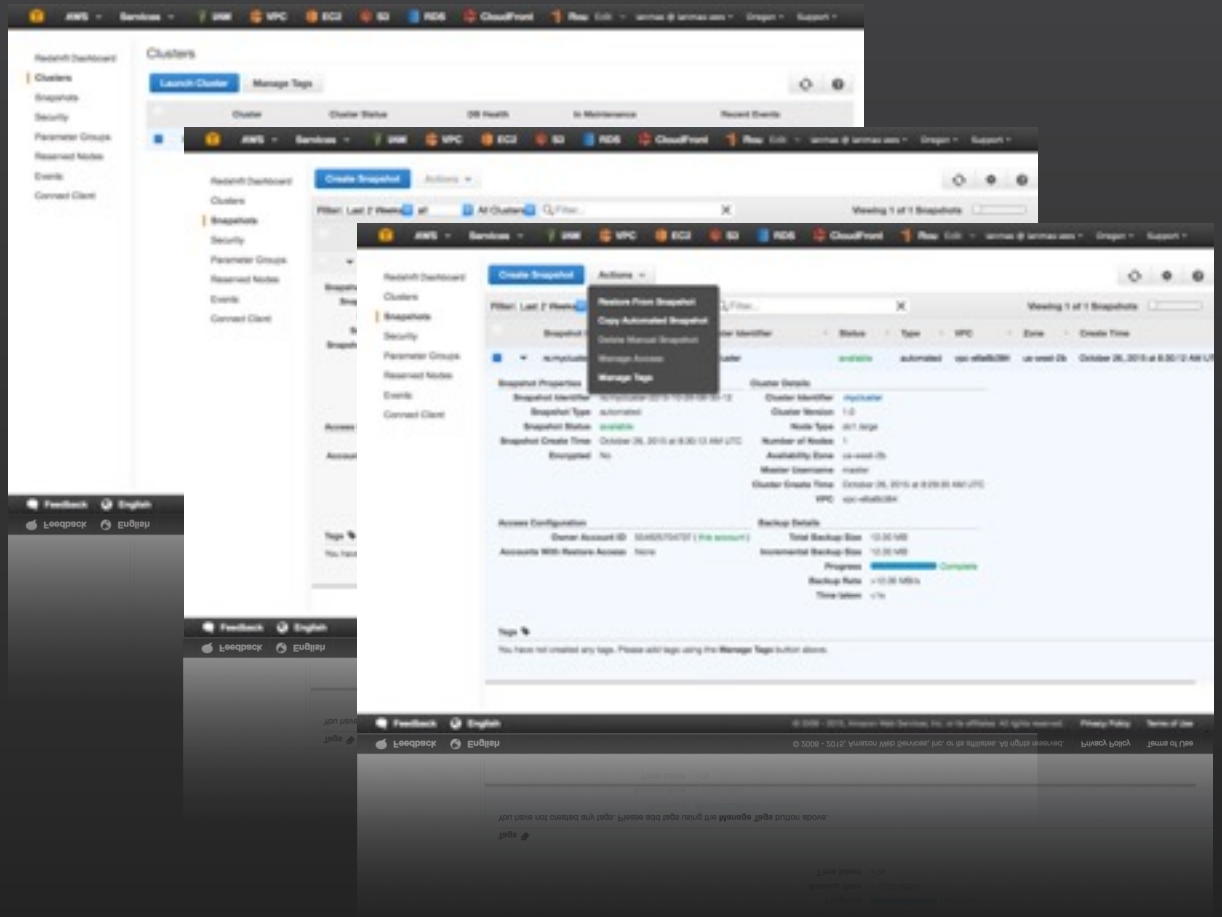
Configurable system snapshot retention period

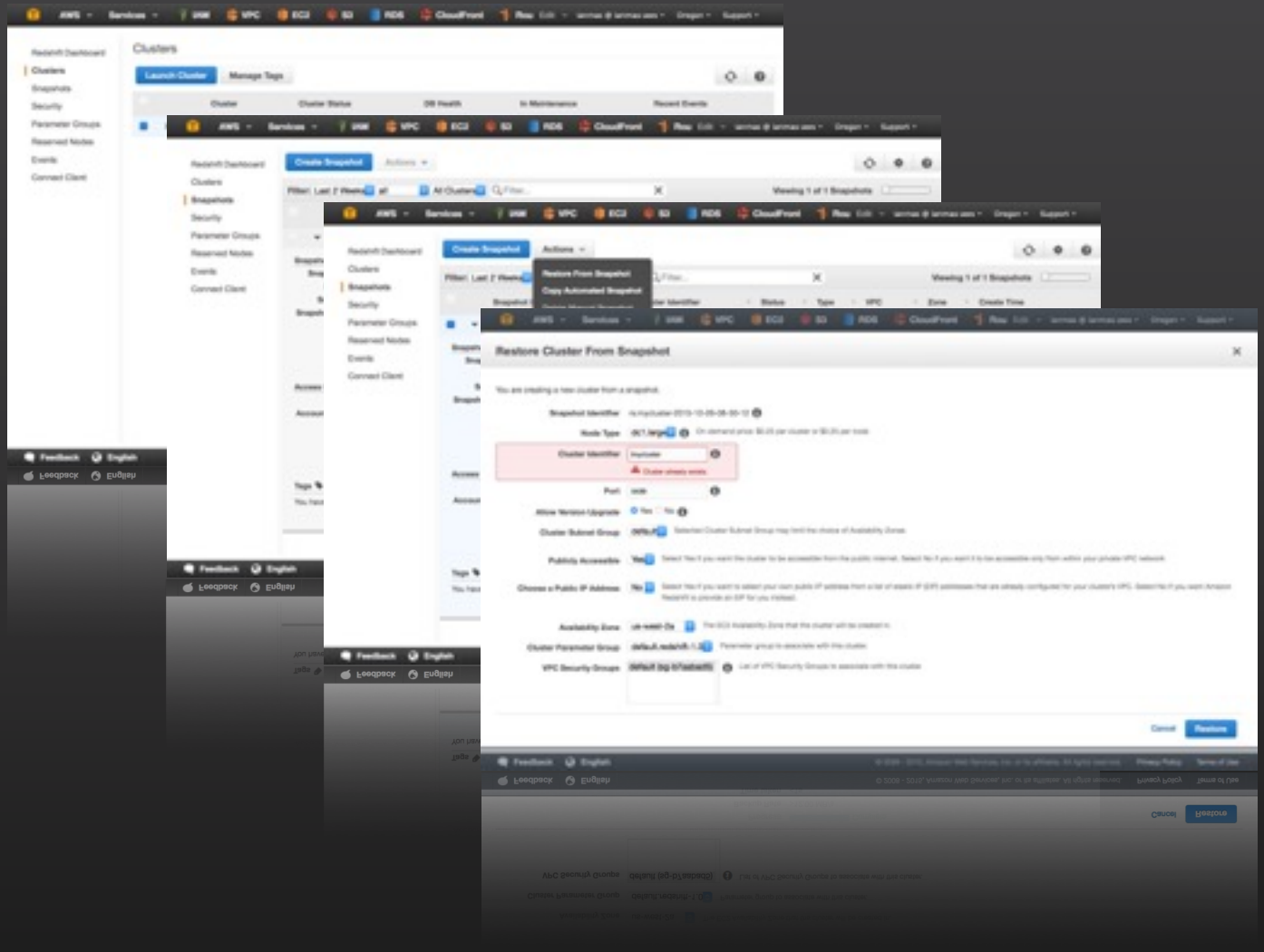
User snapshots on-demand

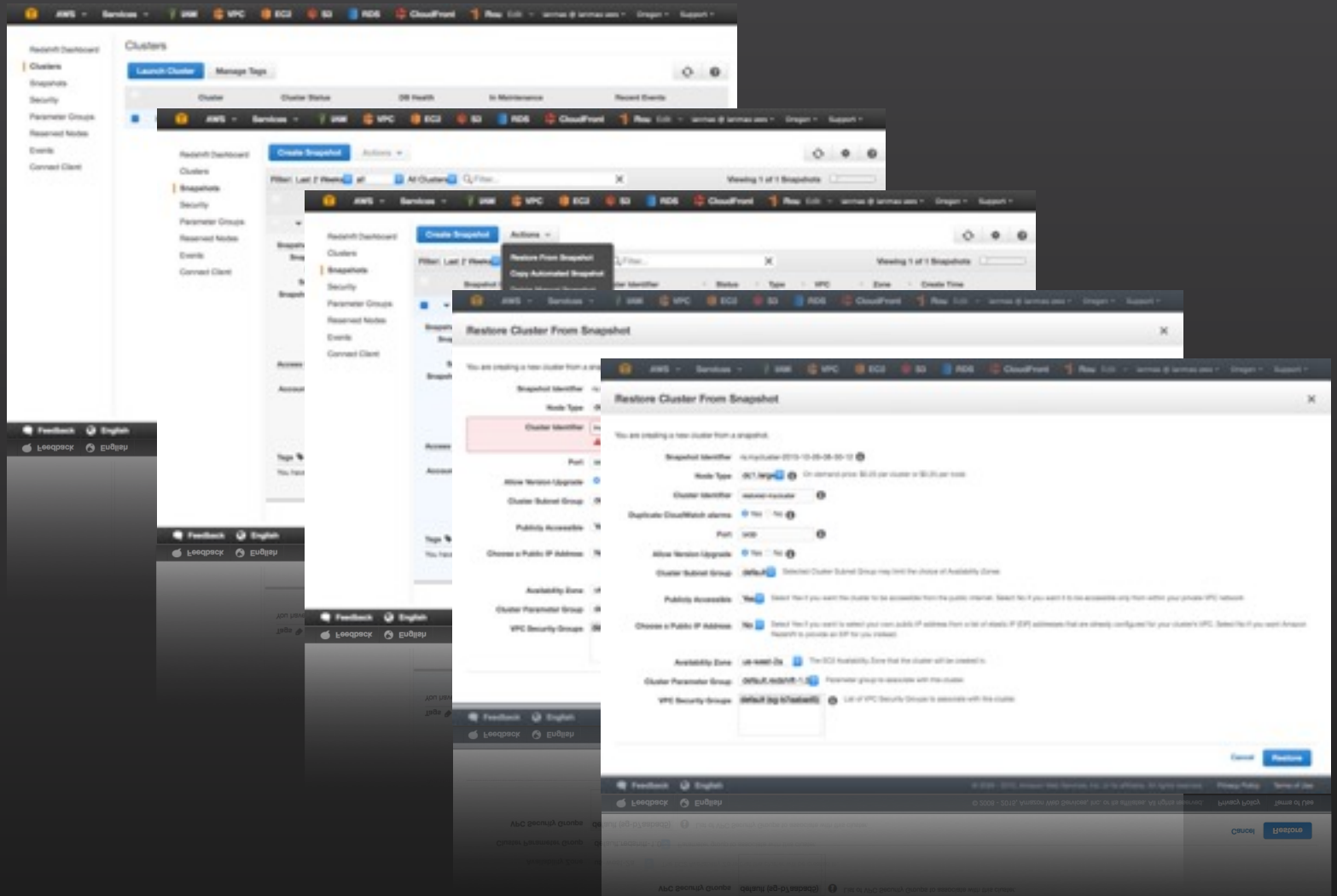
Streaming restores enable you to resume querying faster

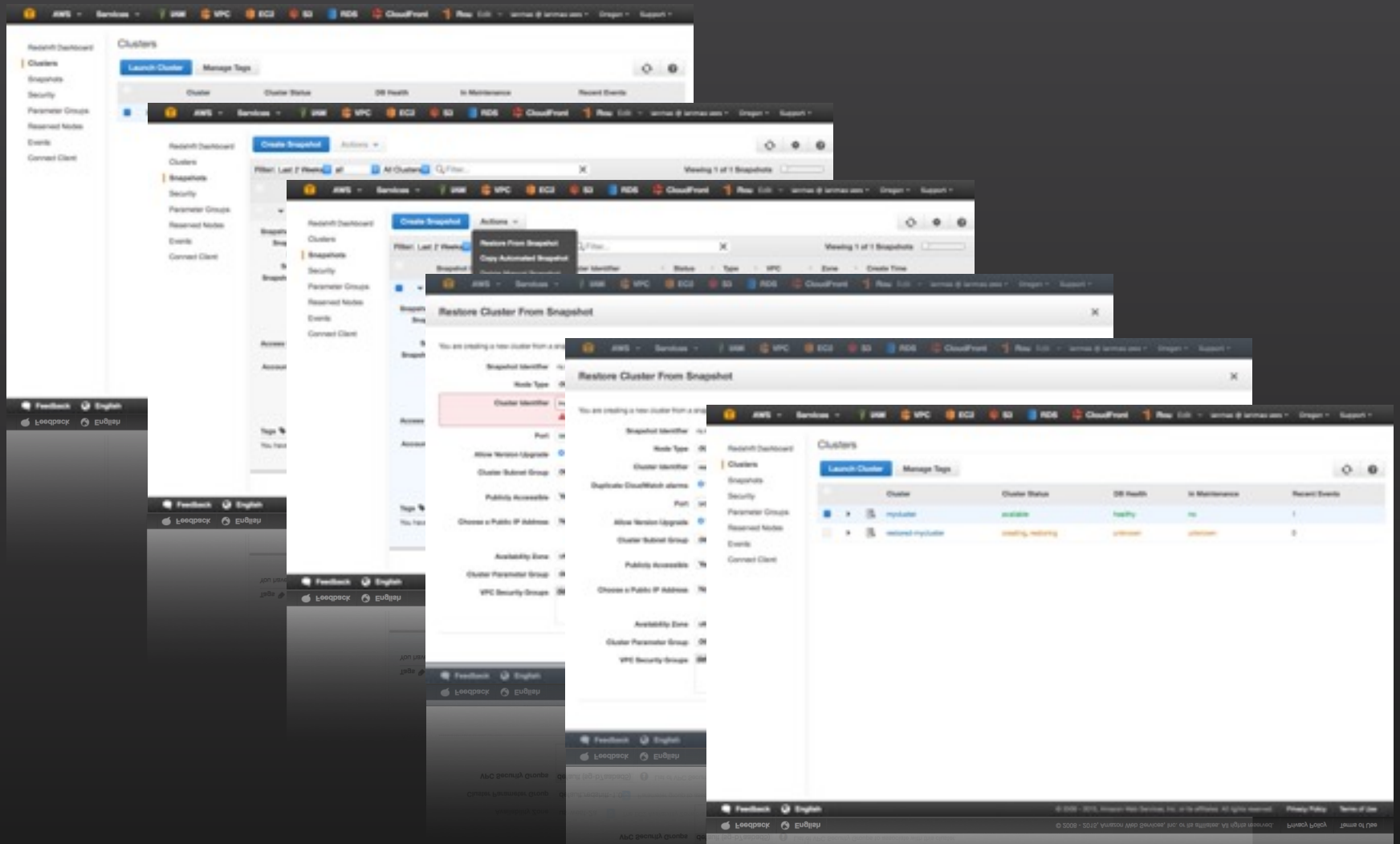

Clusters
Launch Cluster
Manage Tags
Cluster
Cluster Status
DB Health
In Maintenance
Recent Events
mycluster
available
healthy
no
1


Clusters
Launch Cluster
Manage Tags
Create Snapshot
Actions
Snapshots
Feedback
English


Clusters
Launch Cluster
Manage Tags
Create Snapshot
Actions
Restore From Snapshot
Copy Automated Snapshot
Delete Manual Snapshot
Manage Access
Manage Tags
Feedback
English


Clusters
Launch Cluster
Manage Tags
Create Snapshot
Actions
Restore From Snapshot
Copy Automated Snapshot
Restore Cluster From Snapshot
Cancel
Restore


Clusters
Launch Cluster
Manage Tags
Create Snapshot
Actions
Restore From Snapshot
Copy Automated Snapshot
Restore Cluster From Snapshot
Restore Cluster From Snapshot
Cancel
Restore


Clusters
Launch Cluster
Manage Tags
Create Snapshot
Actions
Restore From Snapshot
Copy Automated Snapshot
Restore Cluster From Snapshot
Snapshots
Security
Parameter Groups
Reserved Nodes
Events
Connect Client

CLUSTER DATA REPLICATION

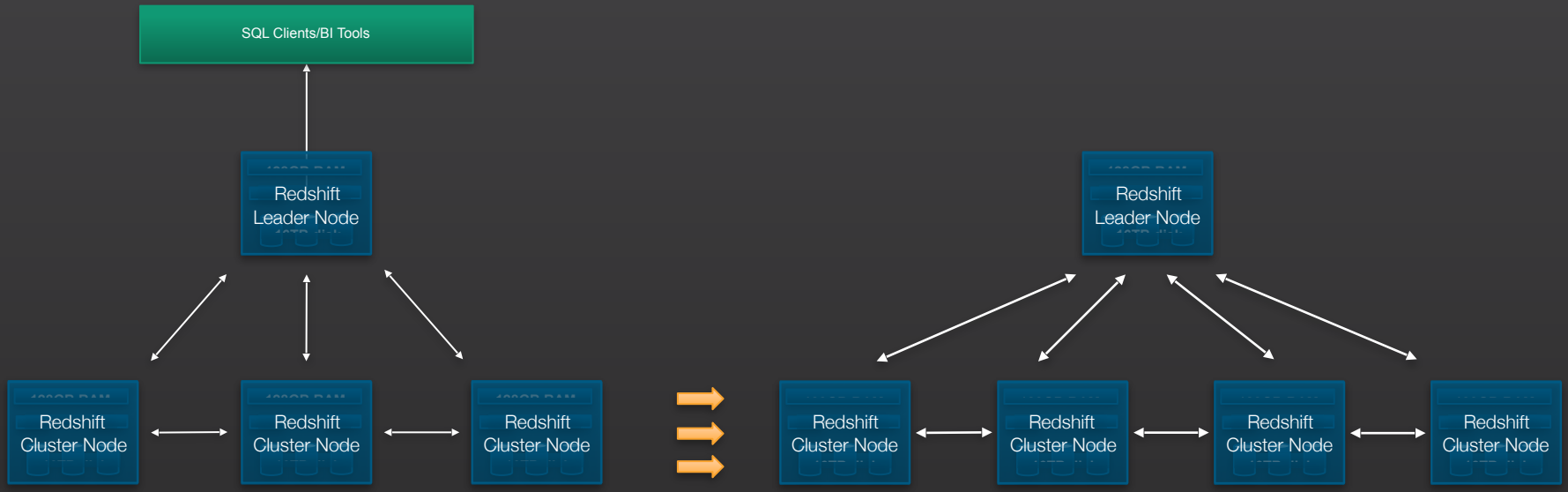
+

AUTOMATED BACKUPS ON AMAZON S3

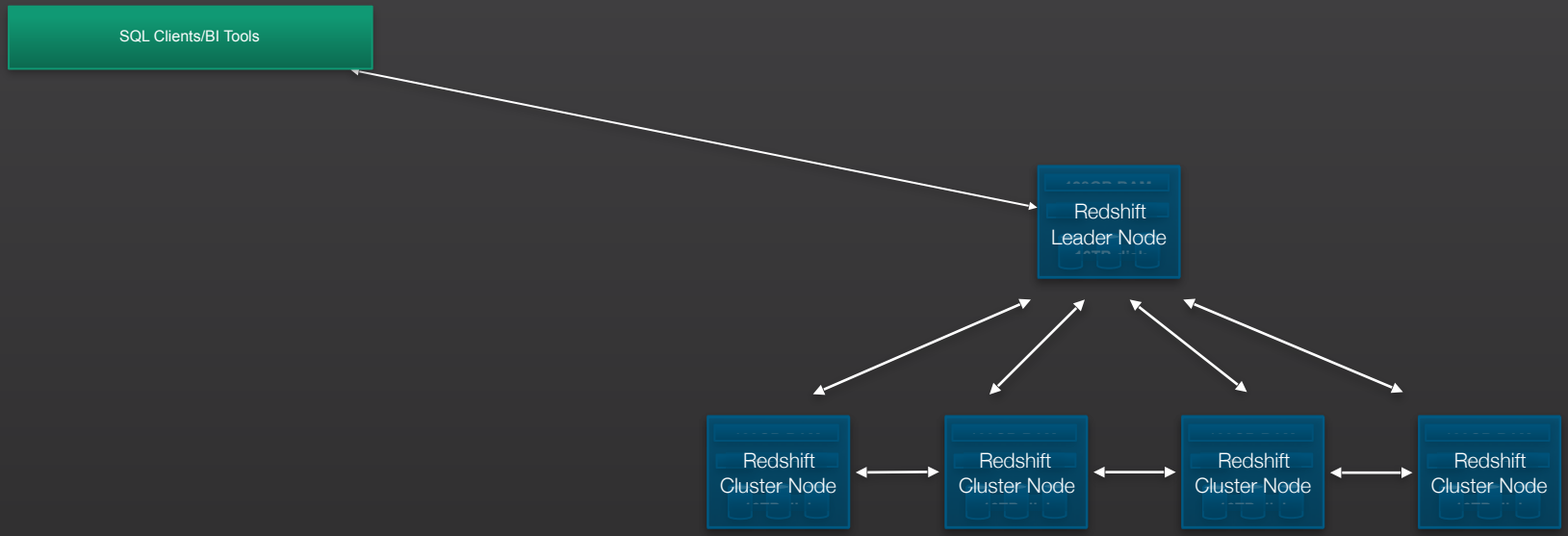
+

NODE MONITORING

UPGRADING & SCALING

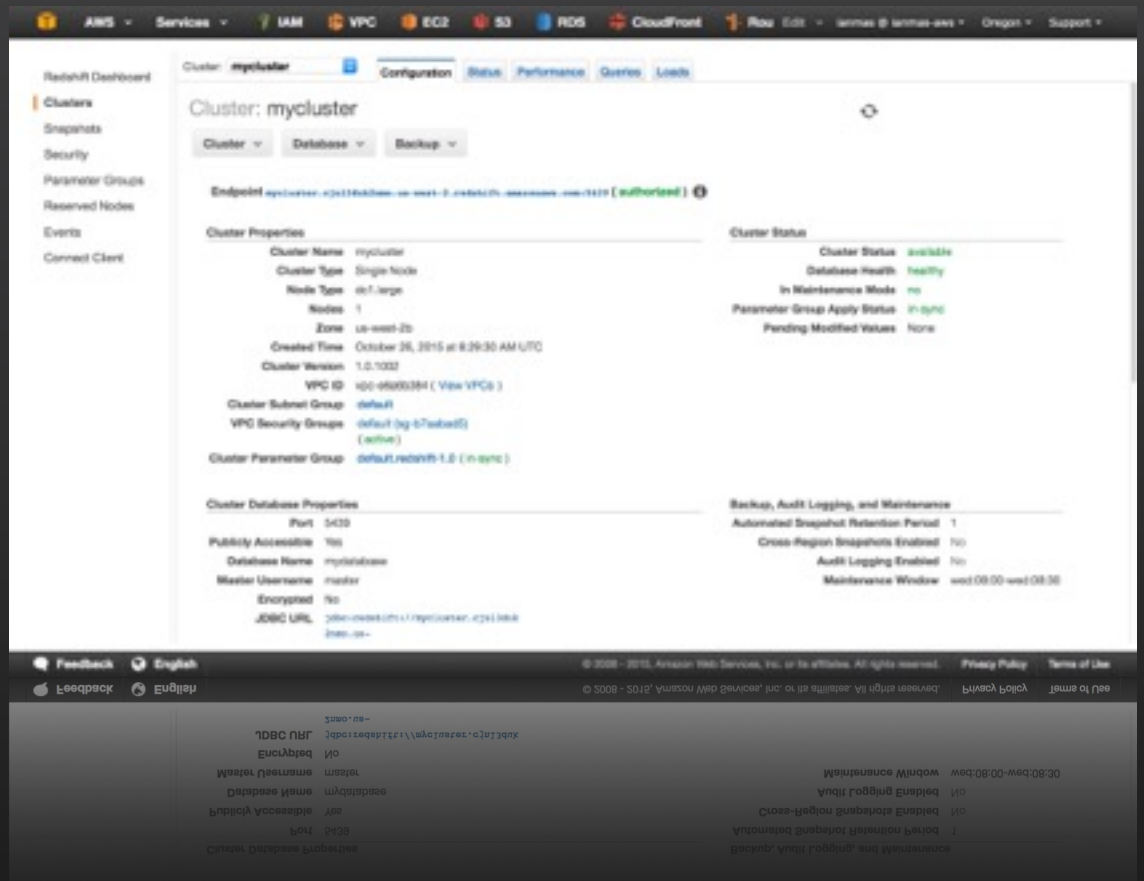


Resize while remaining online
Provision a new cluster in the background
Copy data in parallel from node to node
Only charged for source cluster

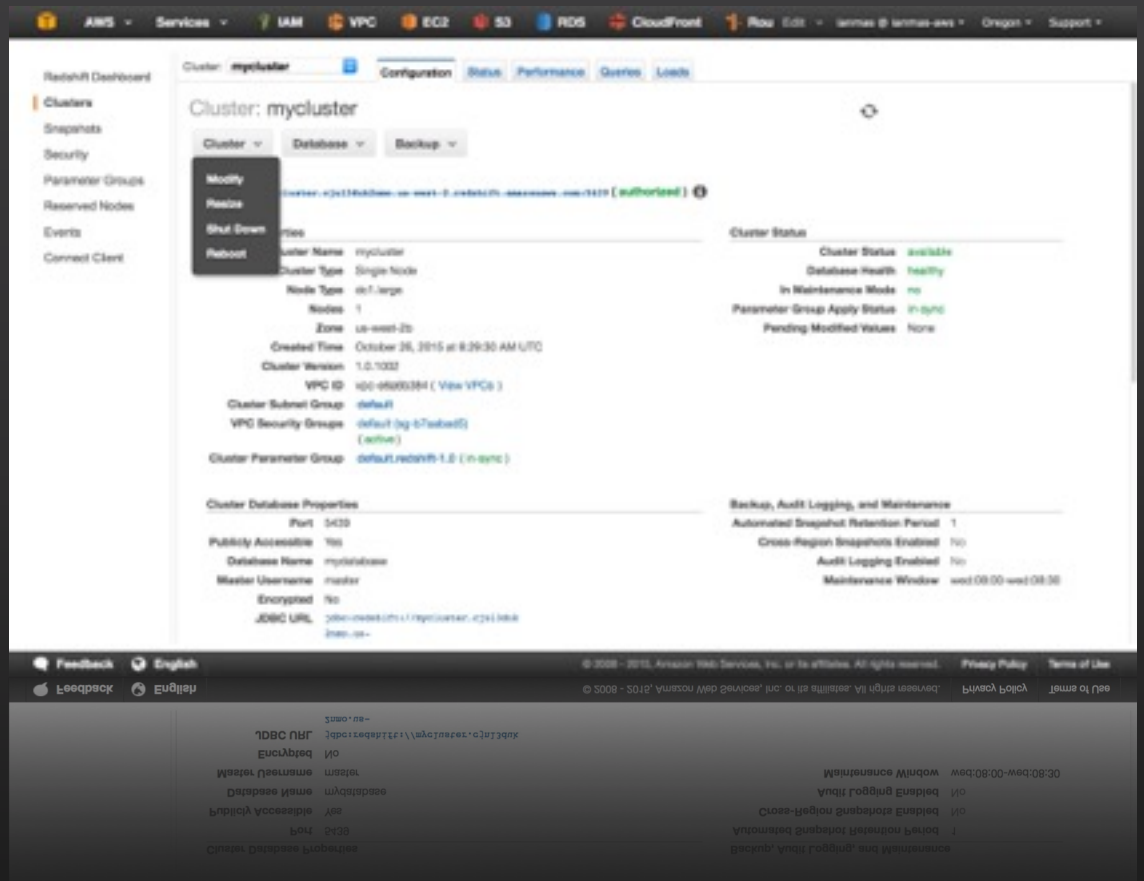


Automatic SQL endpoint switchover via DNS
Decommission the source cluster

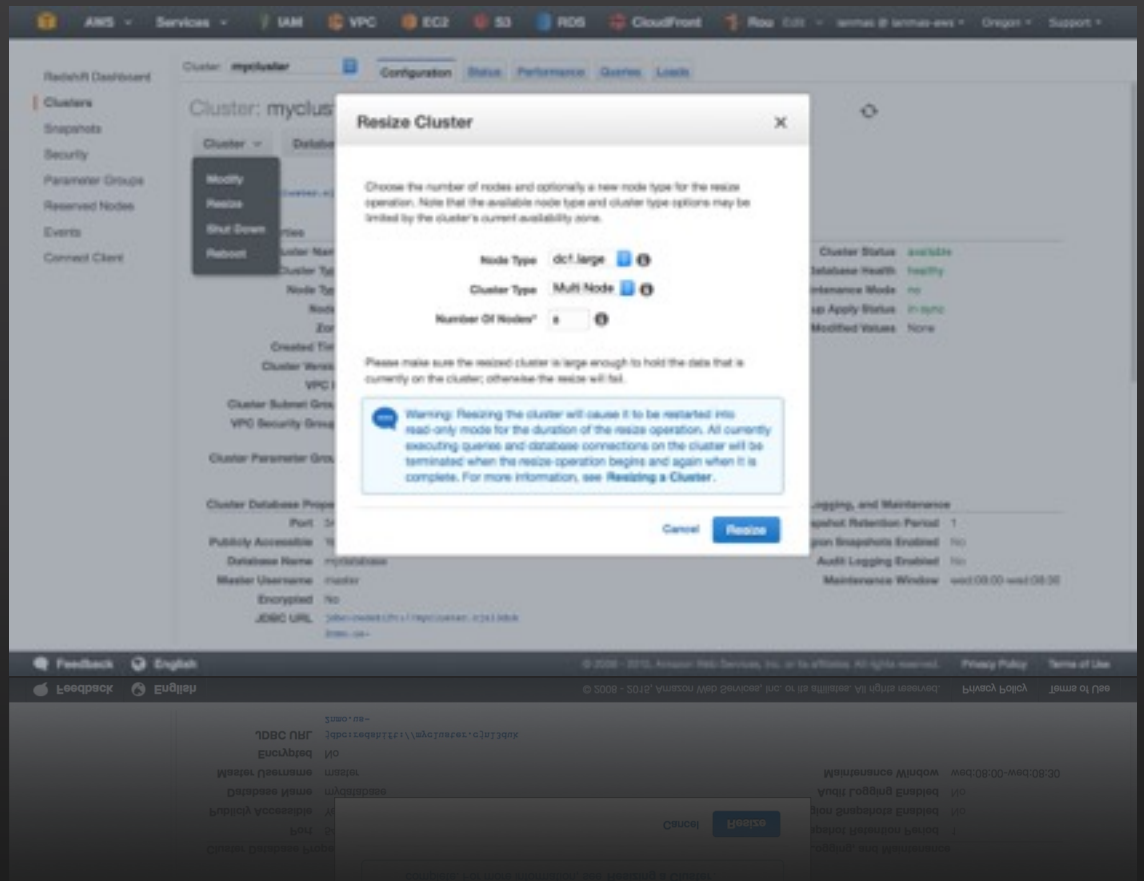
Resizing a Redshift Data Warehouse via the AWS Console



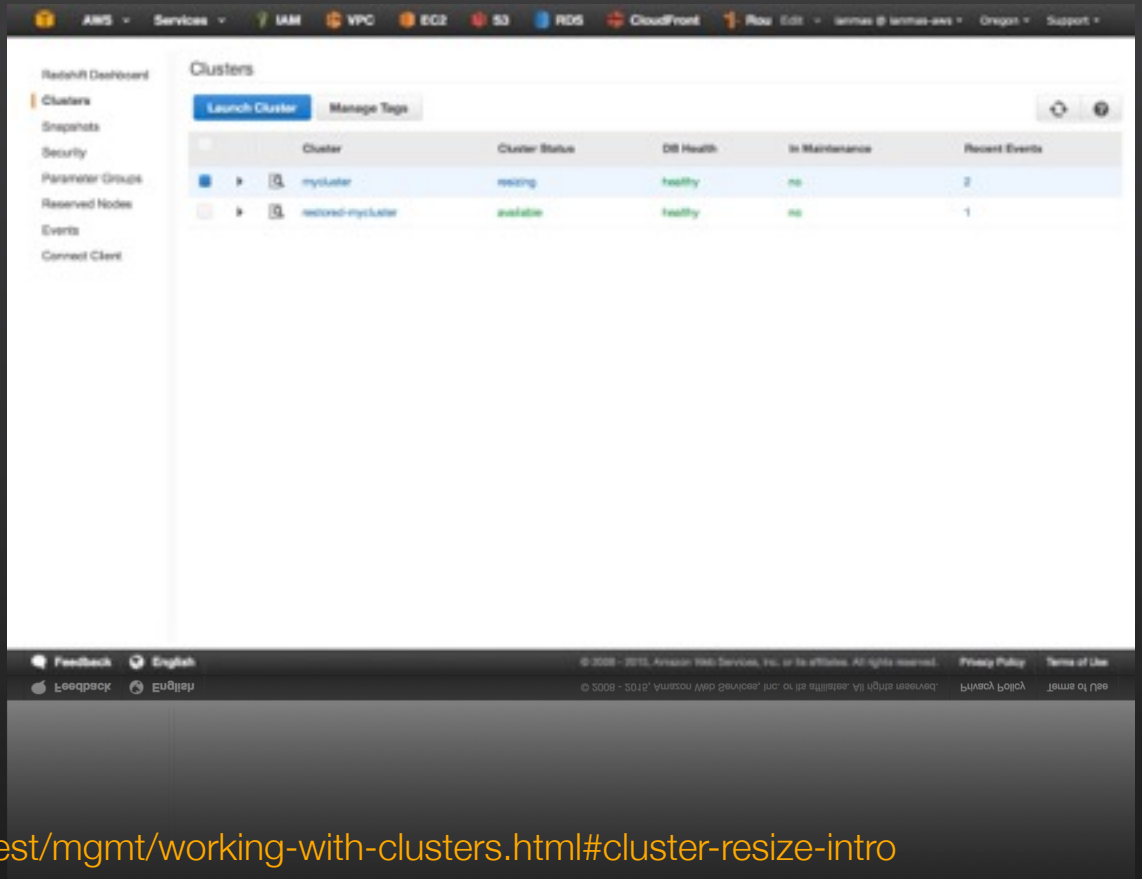
Resizing a Redshift Data Warehouse via the AWS Console



Resizing a Redshift Data Warehouse via the AWS Console



Resizing a Redshift Data Warehouse via the AWS Console



http://docs.aws.amazon.com/redshift/latest/mgmt/working-with-clusters.html#cluster-resize-intro

RECENT FEATURES

Scalar User Defined Functions

You can write UDFs using Python 2.7
- Syntax is largely identical to PostgreSQL UDF syntax
- System & network calls within UDFs are prohibited

Comes with Pandas, NumPy, and SciPy pre-installed

You'll also be able import your own libraries for even more flexibility

https://aws.amazon.com/blogs/aws/category/amazon-redshift/

AWS Official Blog

User Defined Functions for Amazon Redshift

by Jeff Barr | on 11 SEP 2015 | in Amazon Redshift | Permalink | Comments

The Amazon Redshift team is on a tear. They are listening to customer feedback and rolling out new features all the time! Below you will find an announcement of another powerful and highly anticipated new feature.

— Jeff;

Amazon Redshift makes it easy to launch a petabyte-scale data warehouse. For less than $1,000/Terabyte/year, you can focus on your analytics, while Amazon Redshift manages the infrastructure for you. Amazon Redshift's price and performance has allowed customers to unlock diverse analytical use cases to help them understand their business. As you can see from blog posts by Yelp, Amplitude and Cake, our customers are constantly pushing the boundaries of what's possible with data warehousing at scale.

To extend Amazon Redshift's capabilities even further and make it easier for our customers to drive new insights, I am happy to announce that Amazon Redshift has added scalar user-defined functions (UDFs). Using PostgreSQL syntax, you can now create scalar functions in Python 2.7 custom-built for your use case, and execute them in parallel across your cluster.

Here's a template that you can use to create your own functions:

```
sql

  CREATE [ OR REPLACE ] FUNCTION f_function_name
  ( [ argument_name arg_type, ... ] )
  RETURNS data_type
  { VOLATILE | STABLE | IMMUTABLE }
  AS $$
    python_program
  $$ LANGUAGE plpythonu;
```

Scalar UDFs return a single result value for each input value, similar to built-in scalar functions such as ROUND and SUBSTRING. Once defined, you can use UDFs in any SQL statement, just as you would use our built-in functions.

In addition to creating your own functions, you can take advantage of thousands of functions available through Python libraries to perform operations not easily expressed in SQL. You can even add custom libraries directly from S3 and the web. Out of the box, Amazon Redshift UDFs come integrated with the Python Standard Library and a number of other libraries, including:

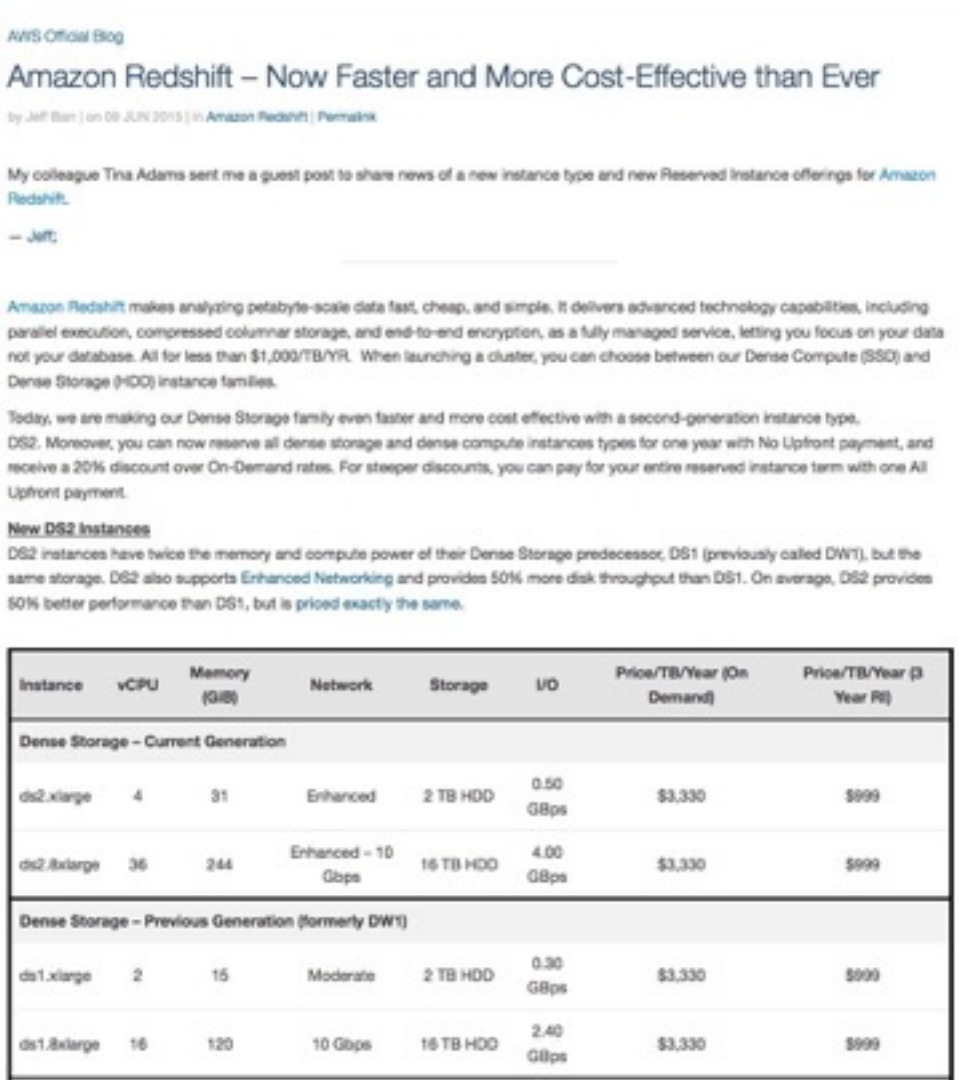
- NumPy and SciPy, which provide mathematical tools you can use to create multi-dimensional objects, do matrix operations, build optimization algorithms, and run statistical analyses.
- Pandas, which offers high level data manipulation tools built on top of NumPy and SciPy, and that enables you to perform data analysis or an end-to-end modeling workflow.
- Dateutil and Pytz, which make it easy to manipulate dates and time zones (such as figuring out how many months are left before the

New DS2 Instances

DS2 instances have twice the memory and compute power of their Dense Storage predecessor, DS1 (previously called DW1), but the same storage.

DS2 also supports Enhanced Networking and provides 50% more disk throughput than DS1.

https://aws.amazon.com/blogs/aws/category/amazon-redshift/



AWS Official Blog

Amazon Redshift – Now Faster and More Cost-Effective than Ever

by Jeff Barr | on 09 JUN 2015 | in Amazon Redshift | Permalink

My colleague Tina Adams sent me a guest post to share news of a new instance type and new Reserved Instance offerings for Amazon Redshift.

— Jeff;

Amazon Redshift makes analyzing petabyte-scale data fast, cheap, and simple. It delivers advanced technology capabilities, including parallel execution, compressed columnar storage, and end-to-end encryption, as a fully managed service, letting you focus on your data not your database. All for less than $1,000/TB/YR. When launching a cluster, you can choose between our Dense Compute (SSD) and Dense Storage (HDD) instance families.

Today, we are making our Dense Storage family even faster and more cost effective with a second-generation instance type, DS2. Moreover, you can now reserve all dense storage and dense compute instances types for one year with No Upfront payment, and receive a 20% discount over On-Demand rates. For steeper discounts, you can pay for your entire reserved instance term with one All Upfront payment.

New DS2 Instances

DS2 instances have twice the memory and compute power of their Dense Storage predecessor, DS1 (previously called DW1), but the same storage. DS2 also supports Enhanced Networking and provides 50% more disk throughput than DS1. On average, DS2 provides 50% better performance than DS1, but is priced exactly the same.

Instance	vCPU	Memory (GiB)	Network	Storage	I/O	Price/TB/Year (On Demand)	Price/TB/Year (3 Year RI)
Dense Storage – Current Generation							
ds2.xlarge	4	31	Enhanced	2 TB HDD	0.50 GBps	$3,330	$999
ds2.8xlarge	36	244	Enhanced – 10 Gbps	16 TB HDD	4.00 GBps	$3,330	$999
Dense Storage – Previous Generation (formerly DW1)							
ds1.xlarge	2	15	Moderate	2 TB HDD	0.30 GBps	$3,330	$999
ds1.8xlarge	16	120	10 Gbps	16 TB HDD	2.40 GBps	$3,330	$999

RESOURCES YOU CAN USE TO LEARN MORE

aws.amazon.com/redshift

Getting Started with Amazon Redshift

aws.amazon.com/redshift/getting-started/

Customers running Analytics in the AWS Cloud

http://aws.amazon.com/solutions/case-studies/analytics/

AWS re:Invent 2015 | Amazon Redshift Deep Dive: Tuning & Best Practices

https://youtu.be/fmy3jCxUliM?list=PLhr1KZpdzukdsblOEVXrCYtvUsDakzYJI

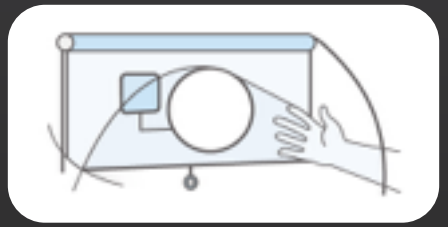
AWS Training & Certification

Self-Paced Labs

Try products, gain new skills, and get hands-on practice working with AWS technologies

aws.amazon.com/training/self-paced-labs

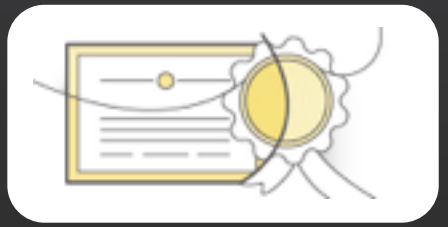
Training



Build technical expertise to design and operate scalable, efficient applications on AWS

aws.amazon.com/training

Certification



Validate your proven skills and expertise with the AWS platform

aws.amazon.com/certification

Follow us for more events & webinars

Ian Massingham — Technical Evangelist

@IanMmmm

@AWS_UKI for local AWS events & news

@AWScloud for Global AWS News & Announcements