

# ドコモのRedshiftの使い方使わせ方

2016/7/22

株式会社 NTTドコモ

サービスイノベーション部 ビッグデータ担当

伊吹 勇郎

- **自己紹介**
- **システムの紹介**
- **課題と対策**
- **まとめ**

## ● 所属/名前

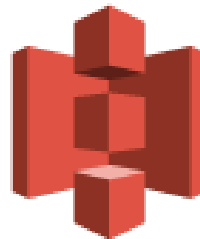
- NTTドコモ R&Dイノベーション本部  
サービスイノベーション部 ビッグデータ担当
- 伊吹勇郎（いぶき たけろう）

## ● 業務

- ビッグデータ統合分析環境 IDAPの開発・運用
  - ・ いっぱい データ あつめる ぶらっとふぉーむ
- ビッグデータ処理基盤の調査・活用推進
  - ・ Apache Spark、Tensorflow...

## ● 好きなAWSサービス

- AMAZON S3

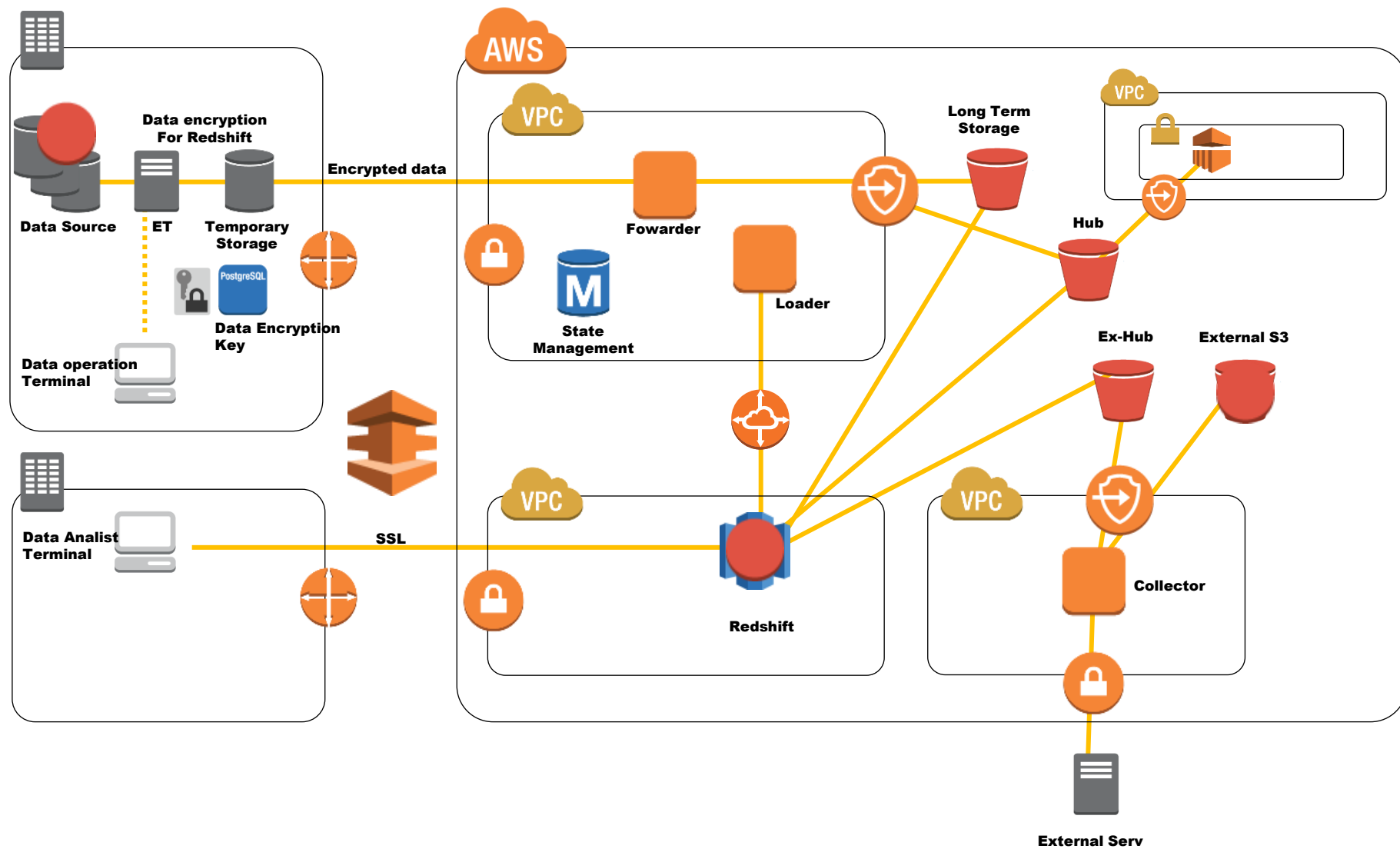


- Redshiftを大規模採用した、  
社内外のデータを統合的に分析可能な環境  
(AWS SUMMIT 2014で発表)
- 社員でDevOpsを実施
  - Data Centerの漏電対策からDockerの導入まで



- 1 **ds2.8xlarge \* 125nodes = 2PB**
- 2 **ユーザが任意のクエリを  
直接Redshiftに接続して発行する**
  - BIツールは使わない
- 3 **ユーザ数百人、同時実行クエリ数は10程度**
- 4 **テーブル数・レコード数は膨大**

# 構成



## Redshift を分析用データの集計・加工に利用



- ・ データが格納されている
- ・ めちゃくちゃ速く集計できる



- ・ PythonやRを使って分析
- ・ 資料作成



大量データの  
機械学習処理を実行

分析や機械学習処理は **Spark** や **Python** で実施

## 速い！



## Redshift

### On-premise DWH ~2014



## 1 セキュリティ



## 2 ユーザ数増大



- 外部犯対策は当然する。  
その上で**内部犯**を気にする



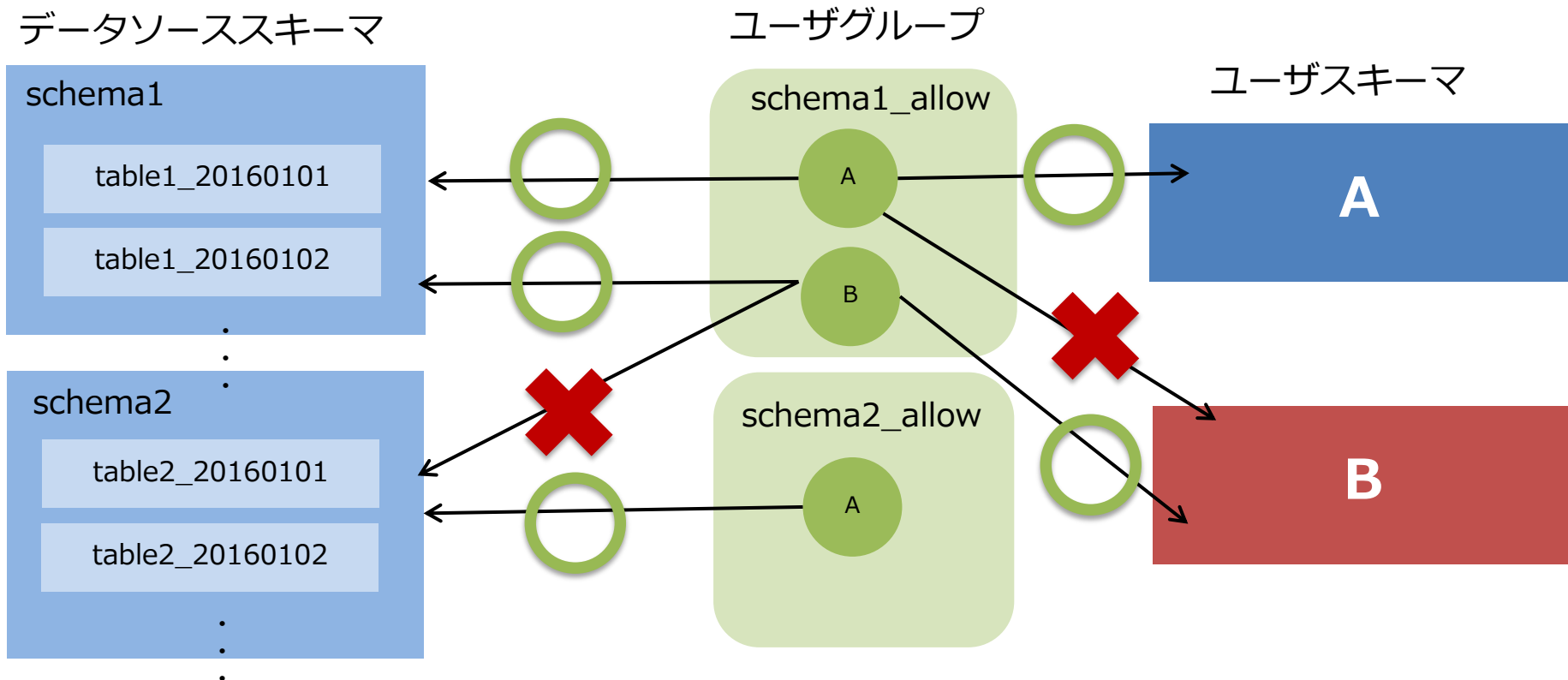
- ユーザは悪さをする（可能性がある）ため、  
システム的に対策をする必要がある
  - 業務上必要がないデータを閲覧する可能性がある
  - 危ないデータを持ち込む・持ち出す可能性がある

- データベースのアクセス権限設定
- システムテーブル権限制御
- UNLOAD制御



# データベースのアクセス権限設定

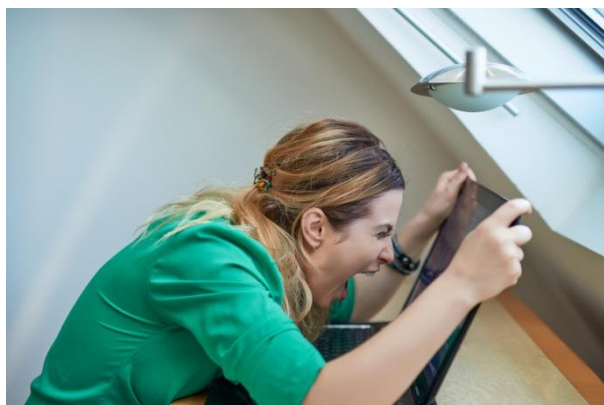
- ユーザには必要なテーブルしか閲覧させない
- ユーザ同士でテーブルの共有はさせない



## テーブル名やその他情報を見せないために システムテーブルの権限を全て **Revoke** !

```
Select distinct 'revoke all on ' || tablename || ' from public '  
from pg_table_def  
where tablename not like '%index'  
and schemaname='pg_catalog';
```

これを実行すると . . . .



¥d できねーじゃねーか ! ! !

システムテーブルを使ったViewを作成し、  
ユーザが自分でテーブル情報等を確認できるようにした

-- **自分**の参照できるテーブル一覧を確認できるSQL  
`select * from tablelist;`

-- **自分**の参照できるテーブルの定義を確認できるSQL  
`select * from tabledeflist;`

-- **自分**の実行クエリ・PID確認  
`select * from querylist;`

-- **自分**が実行できる関数一覧を確認できるSQL  
`select * from functionlist;`

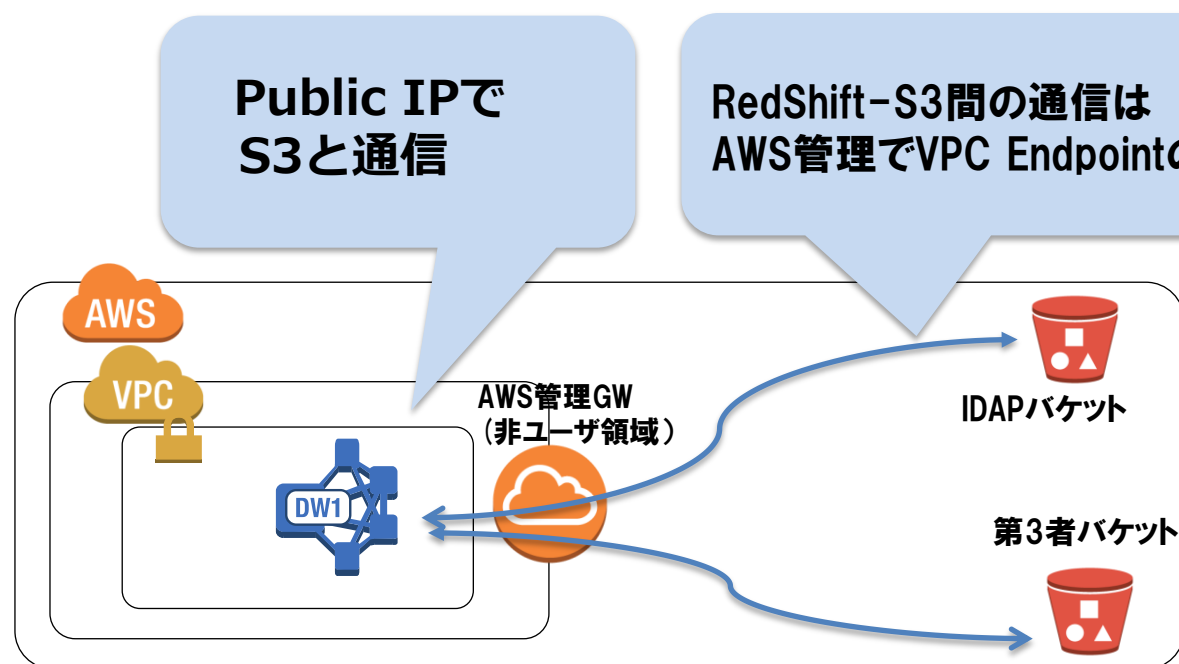
# システムテーブル権限制御

```
create or replace view tablelist as
--- group単位でのテーブル参照
select username, groname as group, nspname as schema, relname as tablename, relkind
from pg_user pu
left join pg_group pg
on ( pg_catalog.array_to_string(pg.grolist,',') like '%' || pu.usesysid || '%' )
left join pg_namespace pn
on
( pg_catalog.array_to_string(pn.nspacl,',') like '%group ' || groname || '=%U%' )
left join pg_class pc
on ( pn.oid = pc.relnamespace and pg_catalog.array_to_string(pc.relacl,',') like '%group ' || groname ||
'=%r%' )
where current_user = username
union
--- 自分のスキーマのテーブル参照(オーナー問わず)
select username, " as group, nspname as schema, relname as tablename, relkind
from pg_user pu
left join pg_namespace pn
on ( nspname = username )
left join pg_class pc
on ( pn.oid = pc.relnamespace )
where current_user = username
union
--- 自分がオーナーのテーブル参照
select username, " as group, nspname as schema, relname as tablename, relkind
from pg_user pu
inner join pg_class pc
on ( pu.usesysid = pc.relowner )
inner join pg_namespace pn
on ( relnamespace = pn.oid )
where current_user = username
order by 1,2,3,4,5;
```

希望があれば  
別途共有します

ユーザにUNLOAD権限を付与すると任意のバケットにデータを出力できるようため、権限を付与しない

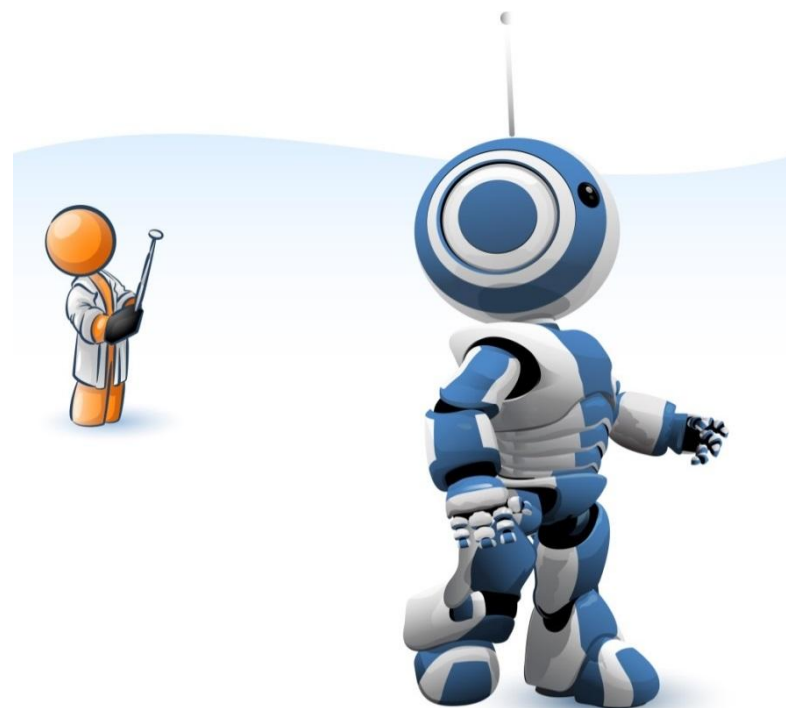
→Redshift上のデータを他サービスで使えない



Redshiftのデータを  
第3者のバケットに  
UNLOADされてしまう  
危険性がある



- 設定ファイルを用意するとRedshiftからデータをUNLOADしてくれるProxyを用意
- EMR用のスクリプトをS3に置くProxyやRedshiftへのデータコピー用Proxyも作成



- テーブル数/スキーマ数枯渇問題
- へたくそクエリ問題
- CTAS問題
- UDF問題

## Redshiftを使い始めて数か月、 ユーザも順調に増えていた所・・・

クラスターごとに作成できるテーブルの最大数(一時テーブルを含む)は 9,900 です。一時テーブルには、ユーザー定義の一時テーブルや、クエリの処理またはシステムメンテナンス中に Amazon Redshift によって作成された一時テーブルが含まれます。ビューはこの上限には含まれません。テーブルの作成の詳細については、「*Amazon Redshift Database Developer Guide*」の「[Create Table Usage Notes](#)」を参照してください。

データベースごとに作成できるスキーマの数は 256 です。スキーマの作成の詳細については、「*Amazon Redshift Database Developer Guide*」の「[Create Schema](#)」を参照してください。

(つ ㇔ ㇔)コ"シコ"シ

(; ° ¨ ¨ °) ...?!

[http://docs.aws.amazon.com/ja\\_jp/redshift/latest/mgmt/amazon-redshift-limits.html](http://docs.aws.amazon.com/ja_jp/redshift/latest/mgmt/amazon-redshift-limits.html)

# テーブル数/スキーマ数枯渇問題

- ユーザにはViewやTemp tableを作成して処理をするように周知
- ユーザ毎のテーブル数を監視し、定期的に注意喚起
- スキーマ数に関しては現時点で対策なし
  - まさか256が限度と思わず・・・



当初はデータサイエンティストのみが使っていたが、  
ユーザ増加につれてだんだん**質**が・・・

→ディスクフルや再起動が発生

Temp tableが消えた・・・



- 長時間実行されているクエリの抽出をしているが根本的な解決になっていない
- 勉強会などを実施し、「よいクエリの書き方」をレクチャしていくしかない



## CREATE TABLE AS SELECT(CTAS)で テーブルを作成する際に圧縮オプションをつけないと、 テーブル容量が肥大化してしまう

```
create table salesusers  
diststyle even  
as  
select sellerid,firstname,lastname,sum(qtysold)  
from  
sales a  
inner join  
users b  
on(a.sellerid=b.userid)  
group by sellerid, firstname, lastname  
order by 4 desc  
;
```

CTASでテーブルを作成すると、  
圧縮オプションが引き継がれなく  
テーブル容量が増えてしまう

## テーブル作成はInsertで実施するように周知しているが 強制することはできない

```
create table salesusers2(  
sellerid integer encode lzo,  
firstname varchar(256) encode lzo,  
lastname varchar(256) encode lzo,  
sum integer encode lzo  
)diststyle even;
```



```
insert into salesusers2(  
select  
sellerid,firstname,lastname,sum(qtysold)  
from  
sales a  
inner join  
users b  
on(a.sellerid=b.userid)  
group by sellerid, firstname, lastname  
order by 4 desc);
```



 **Shop Japan**®

本当はワンダーコアのように  
ダイエット（圧縮オプション適用）  
を自動でさせたい



UDFを実行すると・・・

**Redshiftがリブートするという問題が発生！**

→ デフォルトの最適化オプション設定がVOLATILEで  
負荷が高いことが原因でした

元テーブルの情報が変わらない場合はIMMUTABLEを設定しましょう

– 頻繁に変更がある場合はそもそもRedshiftは適さない

オプション	内容
VOLATILE	実行するたび毎回計算
STABLE	1クエリ内で同じ引数の場合は同じ結果を返す
IMMUTABLE	同一の引数であれば常に同じ結果を返す

- カラムナーデータベースなので、  
必要な列だけ選択するようにしましょう
- メンテナンスウィンドウに注意しましょう  
基本的に週に1回再起動します
- ショートクエリのレイテンシには寛容になりましょう  
10ノード程度なら問題になりませんが、大規模になると影響があります。
- DistkeyとSortkeyの設定をきちんとしましょう
- AWSのSAさんに相談しましょう

## ドコモのビッグデータ統合分析環境IDAPにおけるRedshiftの使い方（使わせ方）をご紹介しました

- Redshiftは125台だと爆速で満足しています
- 次回以降発表してほしい内容ありますか？
  - 全体の設計・セキュリティの話
  - Redshift125台の実力の話
  - 運用の話
  - データフローの話
- ご質問があれば下記までご連絡ください
  - [dcm-cloudconsulting-ml@nttdocomo.com](mailto:dcm-cloudconsulting-ml@nttdocomo.com)

**docomo** cloud package