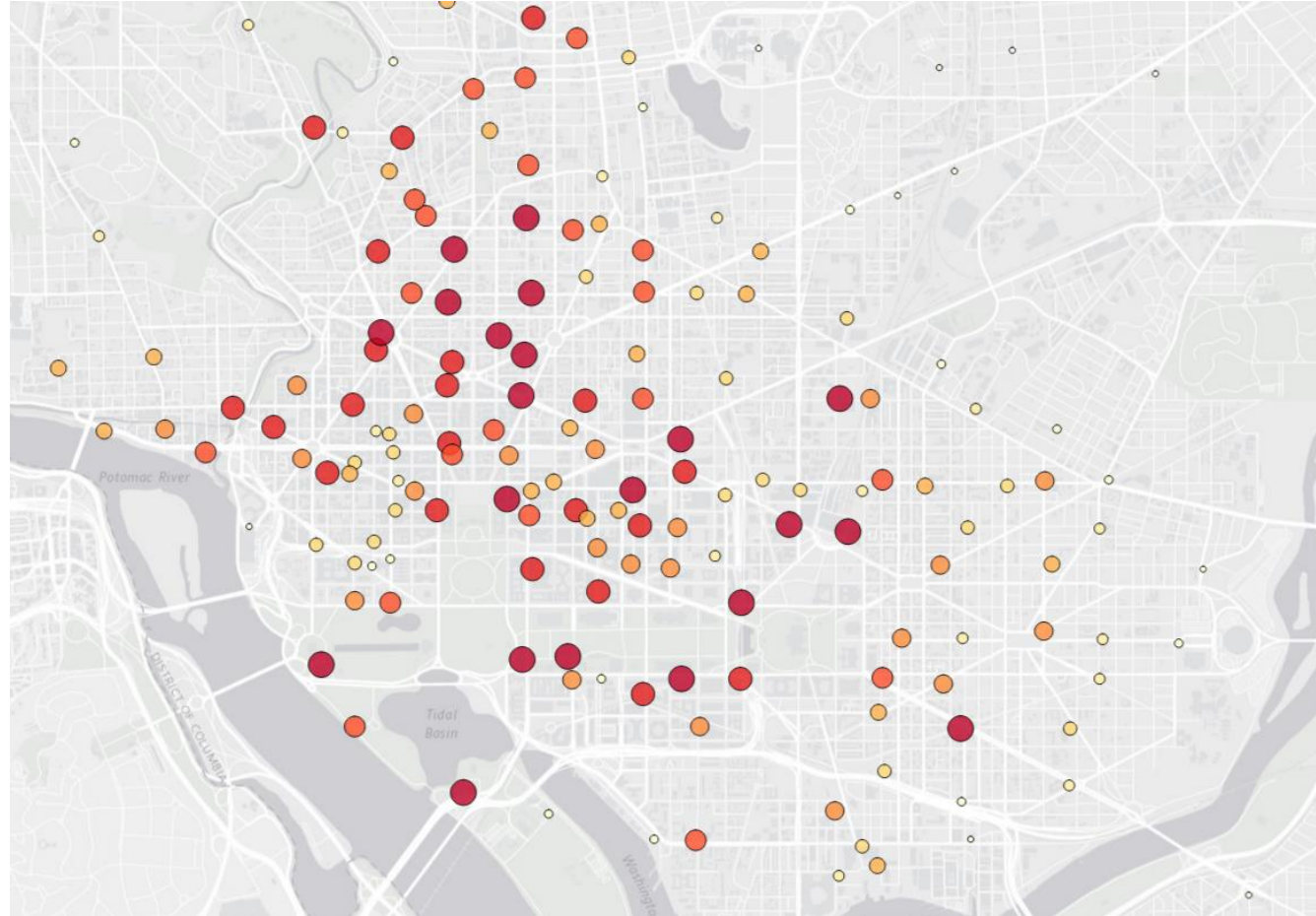


Cabi Visualizations with Amazon Web Services & Leaflet







Anna Petrone @1littlevictory
Transportation Techies Meetup 4/28/2016





Background

Amazon web services offers many cloud-based services:






Compute

-  **EC2**
Virtual Servers in the Cloud
-  **EC2 Container Service**
Run and Manage Docker Containers
-  **Elastic Beanstalk**
Run and Manage Web Apps
-  **Lambda**
Run Code in Response to Events




Storage & Content Delivery

-  **S3**
Scalable Storage in the Cloud
-  **CloudFront**
Global Content Delivery Network
-  **Elastic File System** PREVIEW
Fully Managed File System for EC2
-  **Glacier**
Archive Storage in the Cloud
-  **Import/Export Snowball**
Large Scale Data Transport
-  **Storage Gateway**
Hybrid Storage Integration

Database

-  **RDS**
Managed Relational Database Service
-  **DynamoDB**
Managed NoSQL Database
-  **ElastiCache**
In-Memory Cache
-  **Redshift**
Fast, Simple, Cost-Effective Data Warehousing
-  **DMS**
Managed Database Migration Service






Networking

-  **VPC**
Isolated Cloud Resources
-  **Direct Connect**
Dedicated Network Connection to AWS
-  **Route 53**
Scalable DNS and Domain Name Registration

Developer Tools

-  **CodeCommit**
Store Code in Private Git Repositories
-  **CodeDeploy**
Automate Code Deployments
-  **CodePipeline**
Release Software using Continuous Delivery






Management Tools

-  **CloudWatch**
Monitor Resources and Applications
-  **CloudFormation**
Create and Manage Resources with Templates
-  **CloudTrail**
Track User Activity and API Usage
-  **Config**
Track Resource Inventory and Changes
-  **OpsWorks**
Automate Operations with Chef
-  **Service Catalog**
Create and Use Standardized Products
-  **Trusted Advisor**
Optimize Performance and Security

Security & Identity

-  **Identity & Access Management**
Manage User Access and Encryption Keys
-  **Directory Service**
Host and Manage Active Directory
-  **Inspector** PREVIEW
Analyze Application Security
-  **WAF**
Filter Malicious Web Traffic
-  **Certificate Manager**
Provision, Manage, and Deploy SSL/TLS Certificates


Analytics

-  **EMR**
Managed Hadoop Framework
-  **Data Pipeline**
Orchestration for Data-Driven Workflows
-  **Elasticsearch Service**
Run and Scale Elasticsearch Clusters
-  **Kinesis**
Work with Real-Time Streaming Data
-  **Machine Learning**
Build Smart Applications Quickly and Easily






Internet of Things

-  **AWS IoT**
Connect Devices to the Cloud








Game Development

-  **GameLift**
Deploy and Scale Session-based Multiplayer Games




Mobile Services

-  **Mobile Hub**
Build, Test, and Monitor Mobile Apps
-  **Cognito**
User Identity and App Data Synchronization
-  **Device Farm**
Test Android, FireOS, and iOS Apps on Real Devices in the Cloud
-  **Mobile Analytics**
Collect, View and Export App Analytics
-  **SNS**
Push Notification Service

Application Services

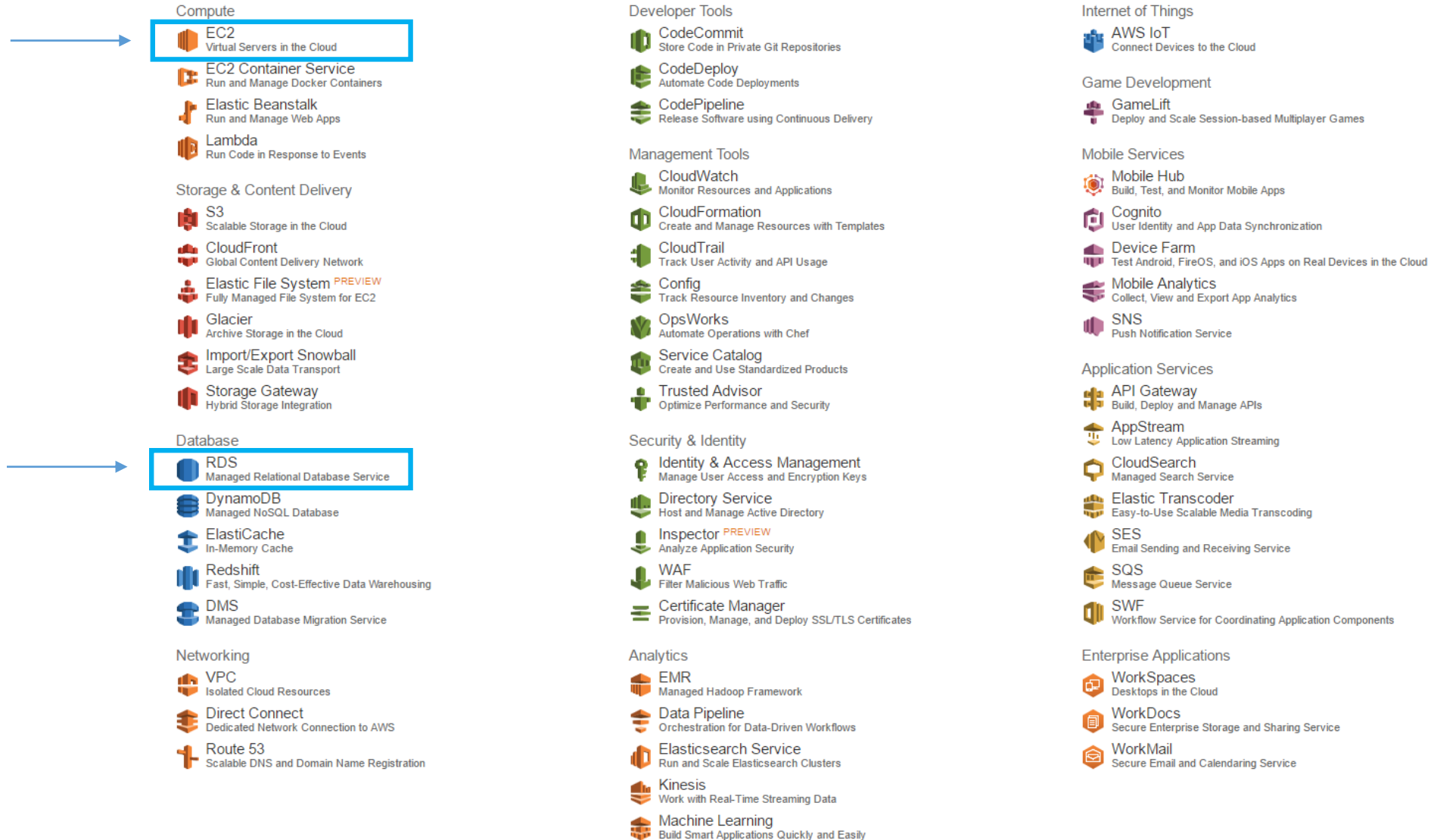
-  **API Gateway**
Build, Deploy and Manage APIs
-  **AppStream**
Low Latency Application Streaming
-  **CloudSearch**
Managed Search Service
-  **Elastic Transcoder**
Easy-to-Use Scalable Media Transcoding
-  **SES**
Email Sending and Receiving Service
-  **SQS**
Message Queue Service
-  **SWF**
Workflow Service for Coordinating Application Components

Enterprise Applications

-  **WorkSpaces**
Desktops in the Cloud
-  **WorkDocs**
Secure Enterprise Storage and Sharing Service
-  **WorkMail**
Secure Email and Calendaring Service

Background

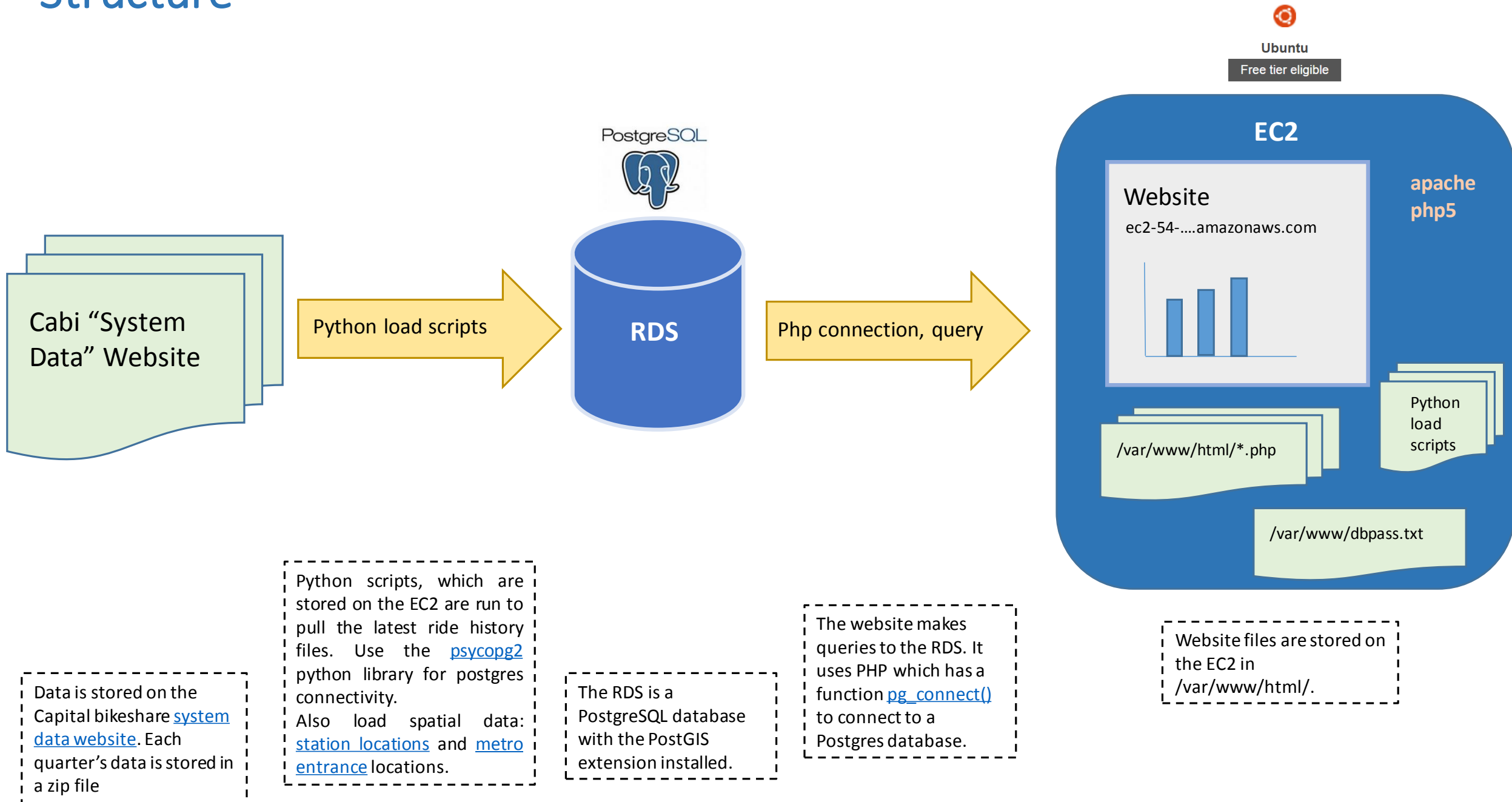
Amazon web services offers many cloud-based services:



EC2 & RDS

- EC2
 - Creates an Ubuntu server in the cloud
 - Ubuntu Server 14.04 LTS (HVM), SSD Volume Type, 64 bit
 - The free tier (“t2.micro”)
 - 1 vCPU, 1GB RAM, 30 GB storage
- RDS
 - Creates a relational database in the cloud
 - PostgreSQL Database
 - Free tier (“t2.micro”)
 - Dev/Test (production option only available for paid)
 - 1 vCPU, 1 GB RAM, 20 GB storage

Structure



AWS Console

- Used to view running instance
- Reboot, modify them if necessary
- Set security permissions (explained on next slide)
- Get database endpoint and EC2 IP address and public DNS

The screenshot displays the AWS Management Console interface for the RDS Dashboard. The top navigation bar includes the AWS logo, 'Services', 'Edit', and user information (Anna Petrone, N. Virginia, Support). The left sidebar lists navigation options: RDS Dashboard, Instances, Clusters, Reserved Purchases, Snapshots, Security Groups, Parameter Groups, Option Groups, Subnet Groups, Events, Event Subscriptions, and Notifications. The main content area shows the 'Launch DB Instance' button, 'Show Monitoring', and 'Instance Actions' buttons. A table lists DB instances, with the selected instance 'cabi' (PostgreSQL engine, available status) highlighted. Below the table, the endpoint is shown as 'cabi.cgzfeinbmbkk.us-east-1.rds.amazonaws.com:5432 (authorized)'. The 'Alarms and Recent Events' section lists four events: 'DB instance restarted' (Apr 28 4:08 PM), 'DB instance shutdown' (Apr 28 4:07 PM), 'DB instance restarted' (Apr 28 12:12 PM), and 'Renamed instance from sample to cabi.' (Apr 28 12:12 PM). The 'Monitoring' section displays metrics for CPU, Memory, Storage, Read IOPS, Write IOPS, and Swap Usage, each with a current value, threshold, and a graph for the last hour.

Alarms and Recent Events

TIME (UTC-4)	EVENT
Apr 28 4:08 PM	DB instance restarted
Apr 28 4:07 PM	DB instance shutdown
Apr 28 12:12 PM	DB instance restarted
Apr 28 12:12 PM	Renamed instance from sample to cabi.

Monitoring

	CURRENT VALUE	THRESHOLD	LAST HOUR
CPU	2.17%	<div><div></div></div>	
Memory	400 MB	<div><div></div></div>	
Storage	9,780 MB	<div><div></div></div>	
Read IOPS	16.1/sec		
Write IOPS	47.6/sec		
Swap Usage	7.45 MB		

Security

Security is handled through “security groups”

- The EC2 permissions:

- Accepts SSH requests from
 - Your IP
- Accepts HTTP requests from
 - Any IP

SSH requests in order to
execute python load scripts



HTTP requests because it
hosts the website.

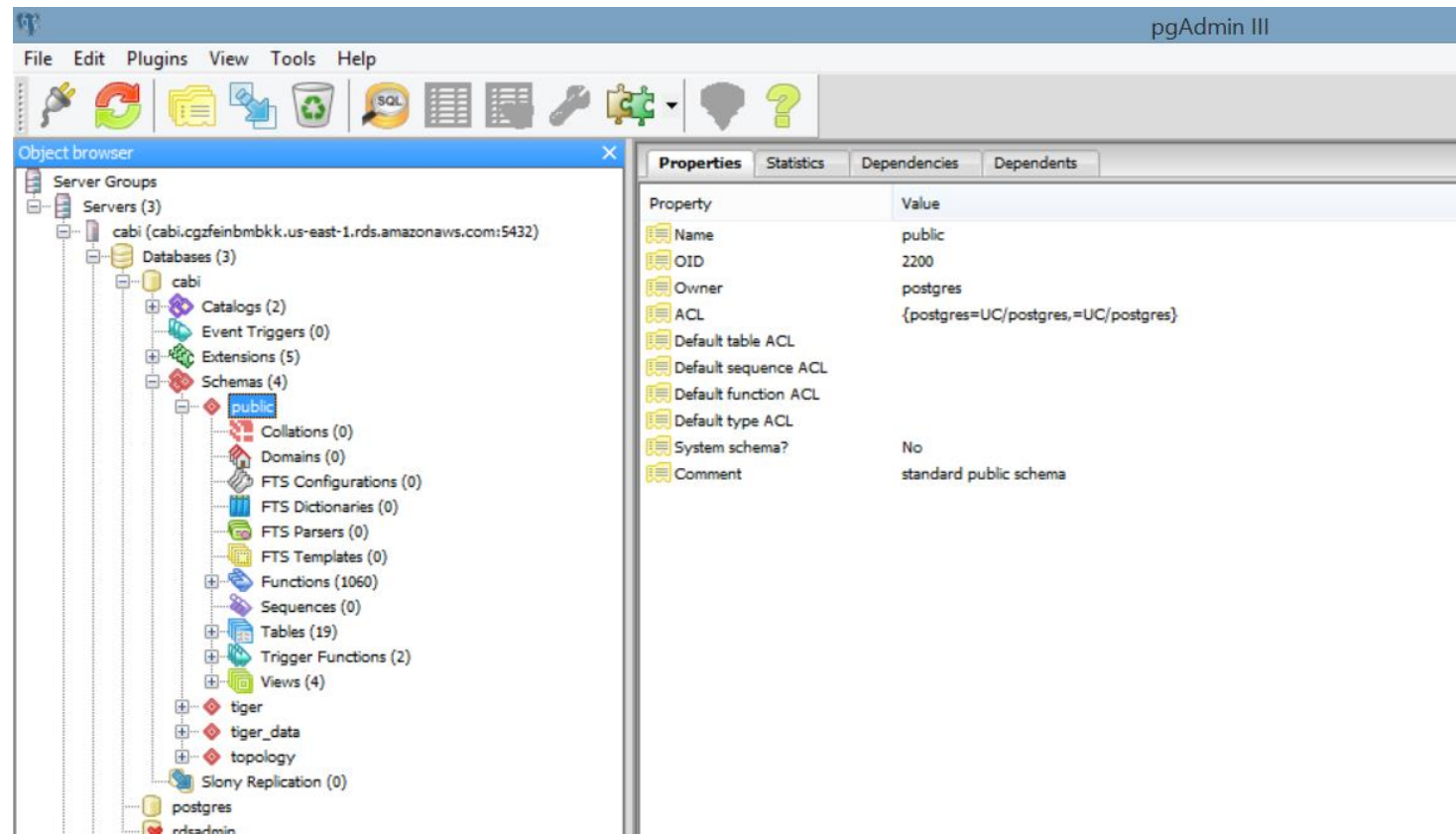
- The RDS permissions:

- Accepts PostgreSQL request from
 - Your IP
 - *The private IP of the EC2*

Accepts Postgres requests
to manually run queries,
and accept website queries

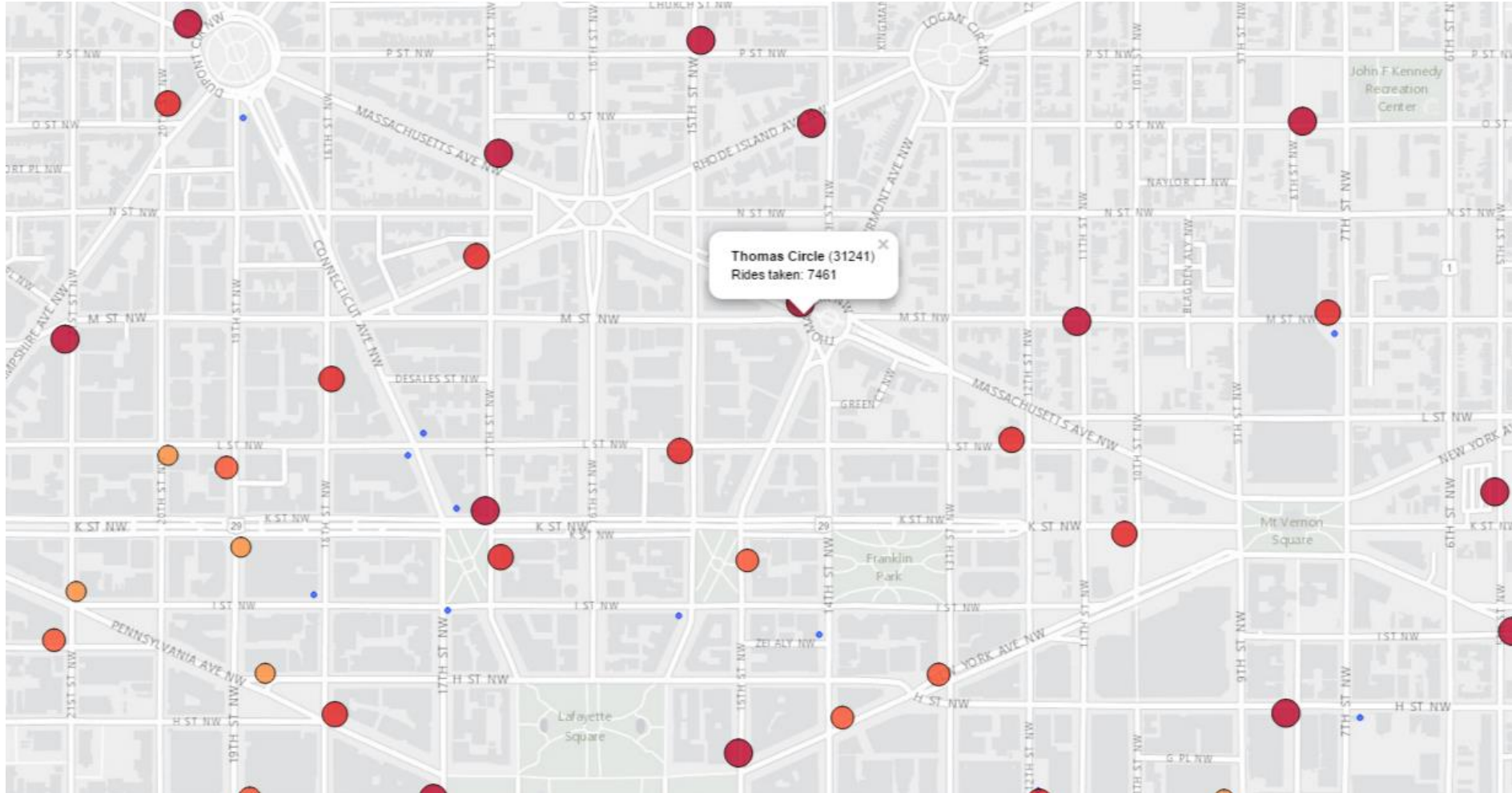
Connecting with pgAdminIII

- <http://www.pgadmin.org/>
- Connect to your DB and run queries against it
- Database must accept PostgreSQL connections from your IP address
- Good for extracting small data sets. Otherwise write a python script using `copy_from`



Live Demo!

- <http://ec2-52-207-212-11.compute-1.amazonaws.com/>



Note about data cleaning

- The rides data come from the [Cabi Website](#) – one per each quarter since 2010Q4
- The formatting is not the same across all files 😭
- See [headers.csv](#) to see the differences
- Most importantly, terminal ID is not present in most files
- Sometimes it's concatenated with the station name, ie
station_name = '14th & Harvard St NW (31105)'
 - In this case, the terminal ID is parsed out
- Otherwise, station name is tried to match into stations table (sourcing from the live xml feed file)
 - But there may have been a name change or station relocation in the past. Can't do anything about this
- Overall ~95% rides have matched start and end terminal ID's

Code snippets

Website code: read user form selection

```
<?php
    $submitted = $_REQUEST['submitted'] ?: 'No filters';

    if($submitted== 'Apply Settings'){
        $output_type = $_REQUEST['output_type'] ?: 'bubble';

        // if user has applied queries, get the values, where now a null means false.
        // the ?: does a coalesce in php5.3 and higher
        $color_by = $_REQUEST['color_by'] ?: 'trip_origins';
        $start_date = $_REQUEST['start_date'] ?: $date_min;
        $end_date = $_REQUEST['end_date'] ?: $date_max;
        $time_period = $_REQUEST['time_period'] ?: 'all';

        $monday = $_REQUEST['monday'] ?: 0;
        $tuesday = $_REQUEST['tuesday'] ?: 0;
        $wednesday = $_REQUEST['wednesday'] ?: 0;
        $thursday = $_REQUEST['thursday'] ?: 0;
        $friday = $_REQUEST['friday'] ?: 0;
        $saturday = $_REQUEST['saturday'] ?: 0;
        $sunday = $_REQUEST['sunday'] ?: 0;

        $registered = $_REQUEST['registered'] ?: 0;
        $casual = $_REQUEST['casual'] ?: 0;
    };

```

Code snippets

Website code: connect to database

```
$dbconn = pg_connect("dbname='cabi' host=$host port=5432 user='postgres' password=$dbpass connect_timeout=5") ;
```

Create query based on inputs

```
$make_temp_table = "
    create table map_temp
    as
    select ride_terminal_id, sum(n_rides) as n_rides
    from ".$from_table."
    where ride_date between ".$start_date." and ".$end_date."
    and ".$where_time."
    and ".$where_weekday."
    and ".$where_member."
    group by 1;";

$get_data = "SELECT row_to_json(fc) as json_feature_list
FROM ( SELECT 'FeatureCollection' As type, array_to_json(array_agg(f)) As features
FROM (SELECT 'Feature' As type
    , ST_AsGeoJSON(the_geom)::json As geometry
    , row_to_json((SELECT l FROM (SELECT n_rides, terminal_id, station_name) As l
    )) As properties
FROM stations As lg
join map_temp as m
on lg.terminal_id = m.ride_terminal_id::int
) As f ) As fc;
";
```

Code snippets

- Leaflet: styling function for bubble colors

```
function getColor(d) { // check here for colors http://colorbrewer2.org/
  return (
    d > quantiles[10] ? '#800026v' :
    d > quantiles[9] ? '#bd0026' :
    d > quantiles[8] ? '#e31a1c' :
    d > quantiles[7] ? '#fc4e2a' :
    d > quantiles[6] ? '#fd8d3c':
    d > quantiles[5] ? '#feb24c' :
    d > quantiles[4] ? '#fed976' :
    d > quantiles[3] ? '#ffeda0':
    d > quantiles[2] ? '#ffffcc' :
    '#ffffe6' );
};
```

- Adding pop-ups

```
function onEachFeature_point(feature, layer) {
  // bind whatever popUp info you want
  if (feature.properties) { //&& feature.properties.neighborhood_name
    layer.bindPopup(
      '<b>'+feature.properties.station_name+'</b><br>'+
      'Rides taken: '+feature.properties.n_rides+'<br>'+
      'Terminal id: '+feature.properties.terminal_id
    );
  }
};
```

Code snippets

- Adding geometries to map

```
var dat = <?php echo $dat ?>;

var geojson;
geojson = L.geoJson(dat, {
  pointToLayer: function (feature, latlng) {
    return L.circleMarker(latlng, style_point(feature));
  },
  onEachFeature: onEachFeature_point
}).addTo(map);
```

Thank you!

Links

- To visualization tool:
 - <http://ec2-52-207-212-11.compute-1.amazonaws.com/>
- To Github Repo w/ instructions & code
 - <https://github.com/ampetr/aws-leaflet>

Anna Petrone
twitter.com/1littlevictory
anna@split.us