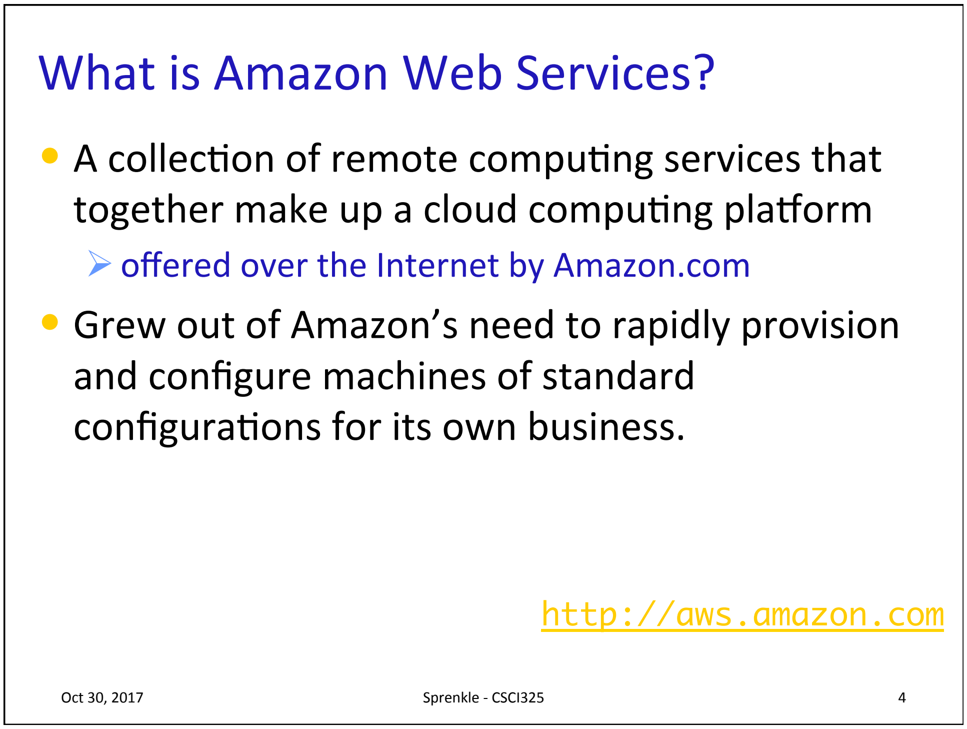
# Today's Objectives

- Phil's Talk Review
- Amazon Web Services
  - Elastic Map Reduce (EMR)

# Phil's Talk

# AMAZON WEB SERVICES (AWS)

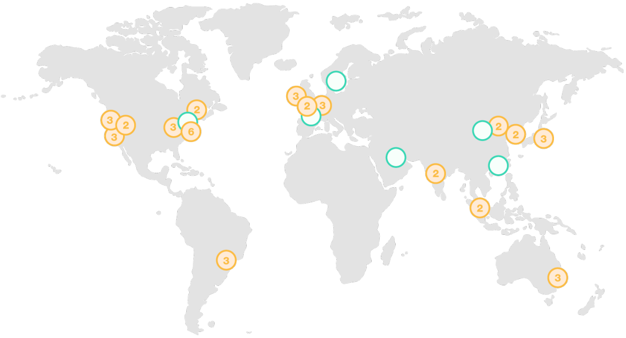


## What is Amazon Web Services?

- A collection of remote computing services that together make up a cloud computing platform
  - offered over the Internet by Amazon.com
- Grew out of Amazon's need to rapidly provision and configure machines of standard configurations for its own business.

http://aws.amazon.com

# Amazon Web Services Architecture

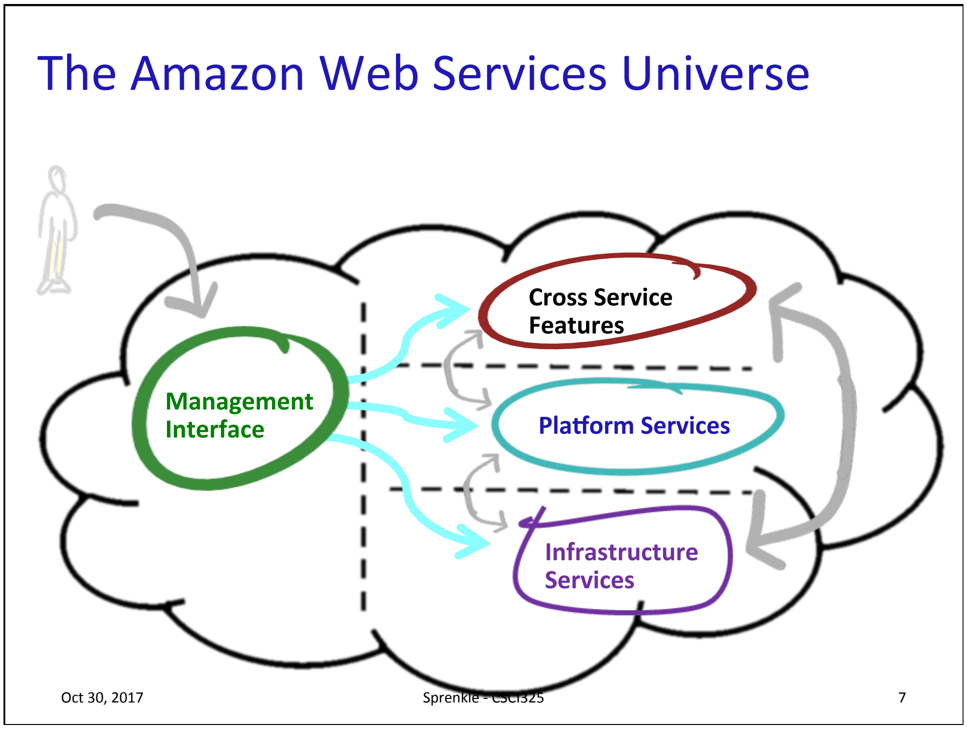


- AWS is located in 16 geographical ***Regions***
  - Region: Geographic location, price, laws, network locality.
  - wholly contained within a single country and all of its data and services stay within the designated Region.
- Each region has multiple ***Availability Zones***
  - distinct data centers providing AWS services
  - isolated from each other to prevent outages from spreading between Zones
  - 44 availability zones

https://aws.amazon.com/about-aws/global-infrastructure/

# Terminology

- Instance: One running virtual machine.
- Instance Type: hardware configuration - cores, memory, disk.
- Instance Store Volume: Temporary disk associated with instance.
- Image (AMI): Stored bits which can be turned into instances.
- Key Pair: Credentials used to access VM from command line.



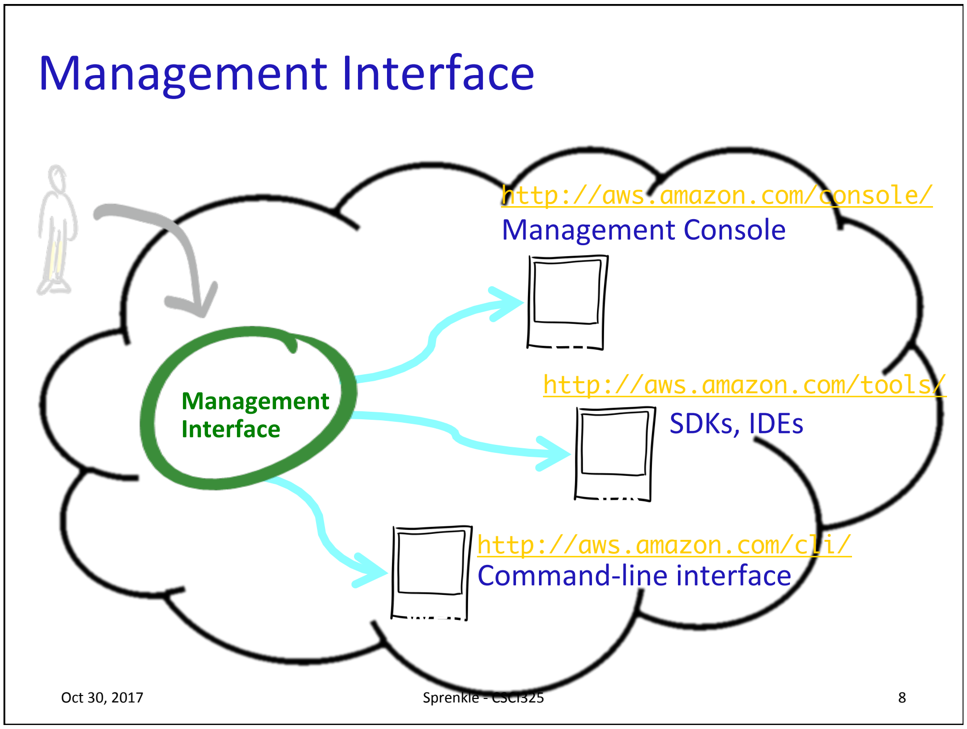
# The Amazon Web Services Universe

Management Interface

Cross Service Features

Platform Services

Infrastructure Services



# Management Interface

Management Interface

http://aws.amazon.com/console/

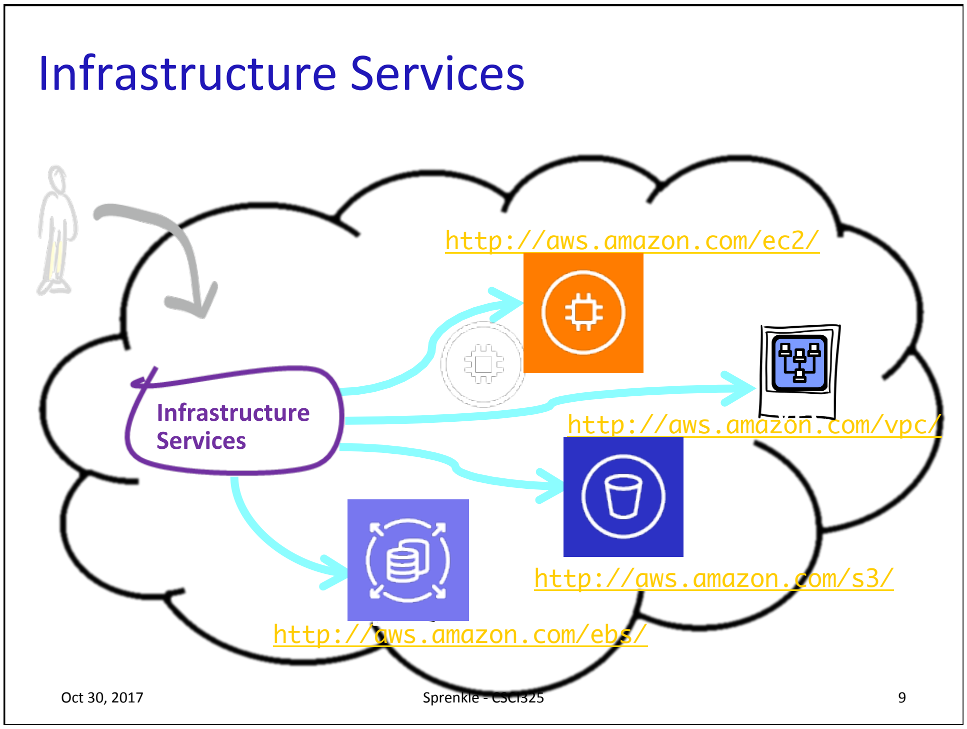
Management Console

http://aws.amazon.com/tools/

SDKs, IDEs

http://aws.amazon.com/cli/

Command-line interface



## Infrastructure Services

Infrastructure Services

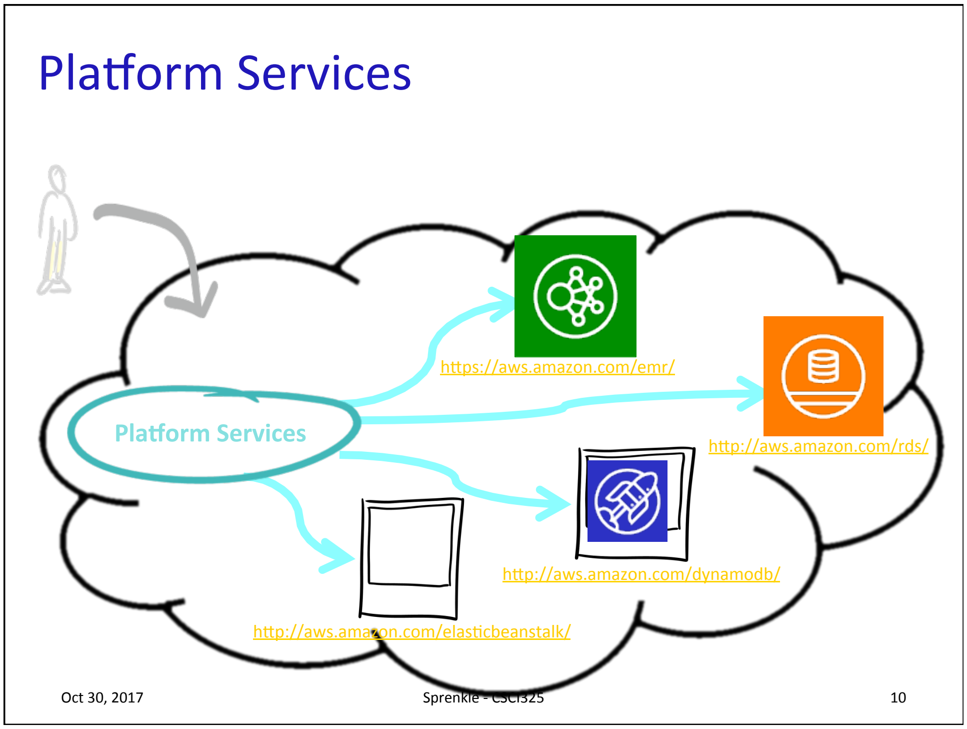
http://aws.amazon.com/ec2/

http://aws.amazon.com/vpc/

http://aws.amazon.com/s3/

http://aws.amazon.com/ebs/

Oct 30, 2017 Sprenkle - CSCI325 9



## Platform Services

Platform Services

https://aws.amazon.com/emr/

http://aws.amazon.com/rds/

http://aws.amazon.com/dynamodb/

http://aws.amazon.com/elasticbeanstalk/

Oct 30, 2017 Sprenkle - CSCI325 10

## Amazon Elastic MapReduce (EMR)

- Web service that makes it easy to quickly and cost-effectively process vast amounts of data using Hadoop
- Distributes data and processing across a resizable cluster of Amazon EC2 instances
- Can launch a *persistent* cluster that stays up indefinitely or a *temporary* cluster that terminates after the analysis is complete
  - Probably want to terminate cluster

## Amazon Elastic MapReduce (EMR)

- Supports a variety of Amazon EC2 instance types and Amazon EC2 pricing options (On-Demand, Reserved, and Spot).
- When launching an Amazon EMR cluster (also called a "job flow"), you choose how many and what type of Amazon EC2 Instances to provision.
- The Amazon EMR price is in addition to the Amazon EC2 price.
- Amazon EMR is used in a variety of applications, including log analysis, web indexing, data warehousing, machine learning, financial analysis, scientific simulation, and bioinformatics.

# WordCount Mapper in Java

```
public static class TokenizerMapper
       extends Mapper<Object, Text, Text, IntWritable> {

    private final static IntWritable one = new IntWritable(1);
    private Text word = new Text();

    public void map(Object key, Text value, Context context)
              throws IOException, InterruptedException {
        StringTokenizer itr = new
StringTokenizer(value.toString());
        while (itr.hasMoreTokens()) {
            word.set(itr.nextToken());
            context.write(word, one);
         }
    }
}
```

# WordCount Reducer in Java

```
public static class IntSumReducer
    extends Reducer<Text, IntWritable, Text, IntWritable> {
    private IntWritable result = new IntWritable();

    public void reduce(Text key, Iterable<IntWritable>
values, Context context)
              throws IOException, InterruptedException {
       int sum = 0;
       for (IntWritable val : values) {
          sum += val.get();
       }
       result.set(sum);
       context.write(key, result);
    }
}
```

## WordCount.java

```
public class WordCount {
   public static void main(String[] args) throws
Exception {
      Configuration conf = new Configuration();
      Job job = Job.getInstance(conf, "word count");
      job.setJarByClass(WordCount.class);
      job.setMapperClass(TokenizerMapper.class);
      job.setCombinerClass(IntSumReducer.class);
      job.setReducerClass(IntSumReducer.class);
      job.setOutputKeyClass(Text.class);
      job.setOutputValueClass(IntWritable.class);
      FileInputFormat.addInputPath(job, new
Path(args[0]));
      FileOutputFormat.setOutputPath(job, new
Path(args[1]));
      System.exit(job.waitForCompletion(true) ? 0 : 1);
   }
}
```

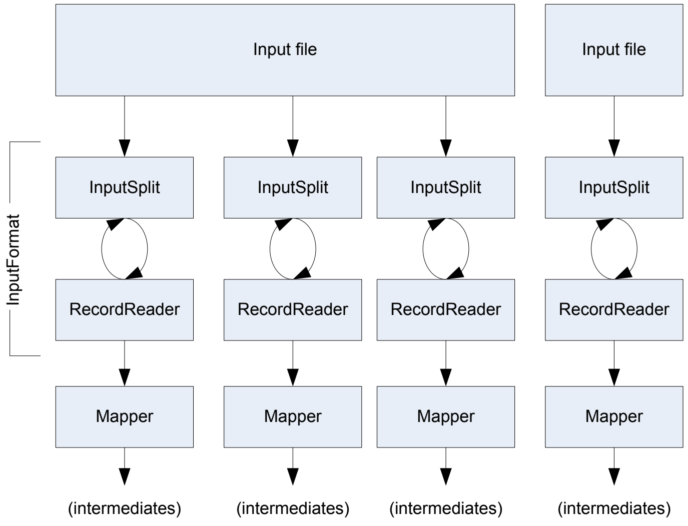
## Nested Classes

- Nested class: member of enclosing class
- Non-static nested classes/inner classes
  - Have access to members of enclosing class, even if private
- Static nested classes do not have access to (instance) members of enclosing class

# Solutions

- Original code given
  - All part of one Java class file
- Alternative:
  - Classes in separate Java class files/not inner classes
  - The way I organized your example code in GitHub so that you may have an easier time with sharing/ collaborating

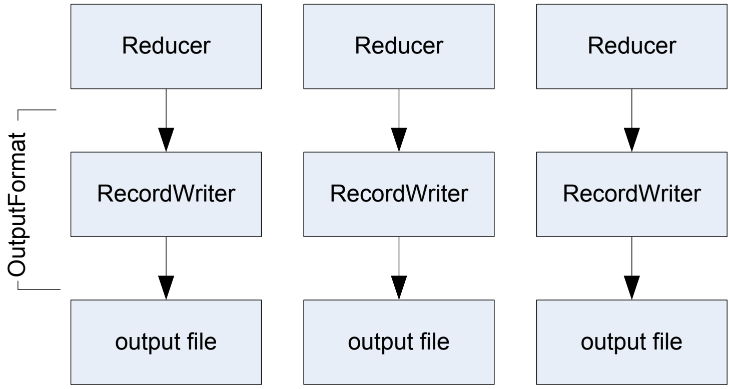
# Getting Data To The Mapper

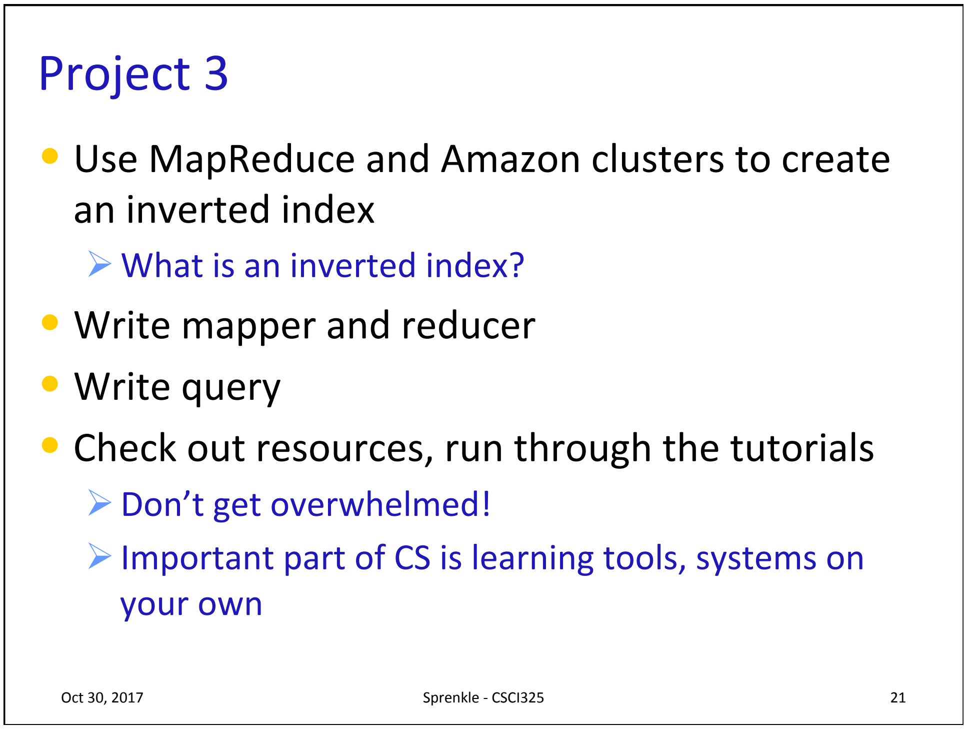


## Mapper<KEYIN,VALUEIN,KEYOUT,VALUEOUT>

- FileInputFormat: Key – offset of data in its file

## Finally: Writing The Output





# Project 3

- Use MapReduce and Amazon clusters to create an inverted index
  - What is an inverted index?
- Write mapper and reducer
- Write query
- Check out resources, run through the tutorials
  - Don’t get overwhelmed!
  - Important part of CS is learning tools, systems on your own