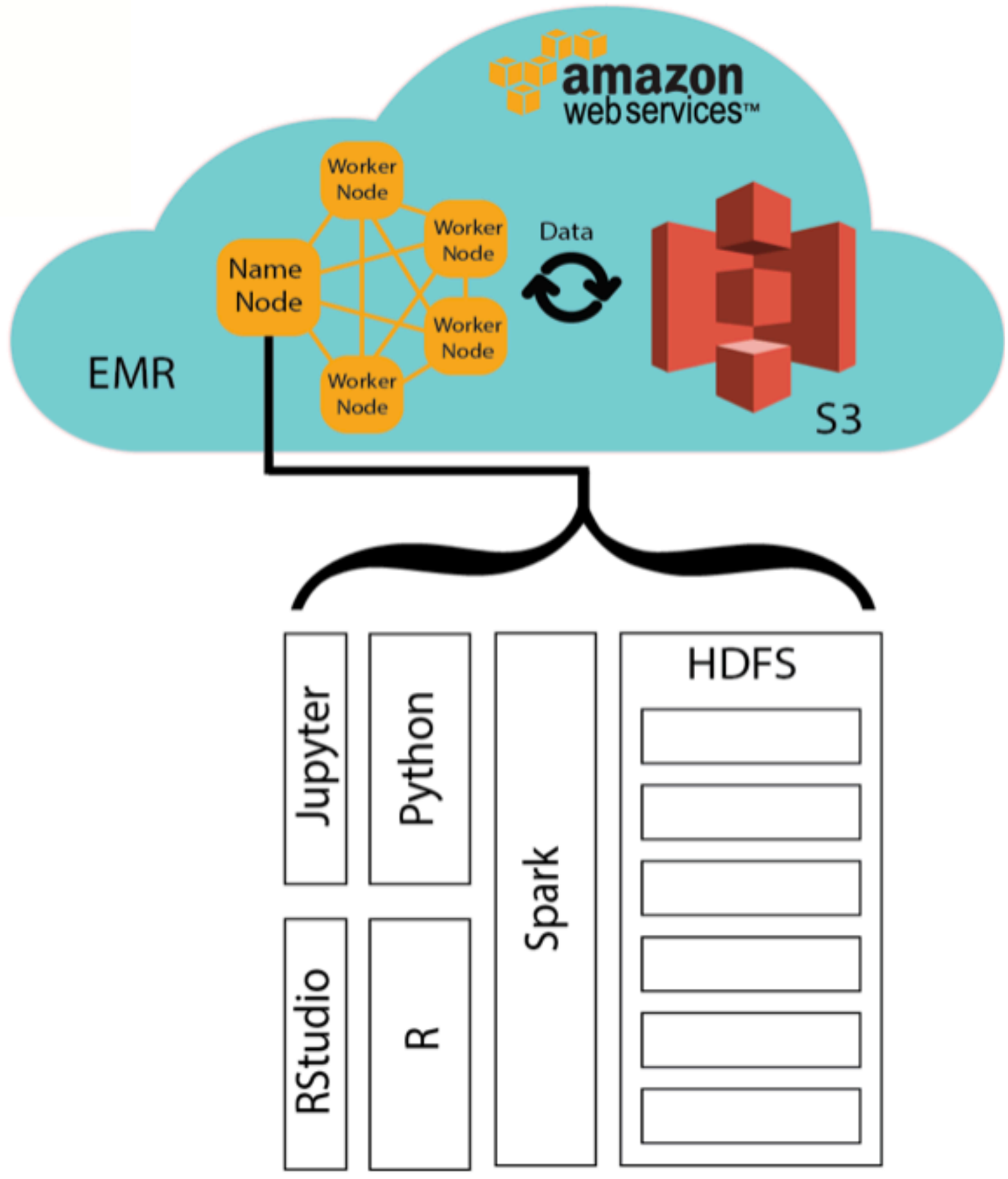
DevOps

A Practical Introduction


amazon
web services™
Worker Node
Worker Node
Worker Node
Worker Node
Name Node
Data
EMR
S3
Jupyter
Python
Spark
HDFS
RStudio
R

Motivating Example

- On every computer, I had to:
  - Create an AWS Account and assign correct permissions
  - Download a PPM Key, install PuttyGen, create a PPK key;
  - Install AWS Command Line
  - Install Putty for Secure Shell
  - Run FoxyProxy Chrome Extension (Port Forwarding)

Motivating Example

Specify Details

Specify a stack name and parameter values. You can use or change the default parameter values, which are defined in the AWS CloudFormation template. Learn more.

Stack name

Parameters

EMR Options

EC2KeyName
SSH key pair to use for EMR node login

VPC Search by ID, or Name tag value
VPC for EMR nodes.

Subnet Search by ID, or Name tag value
Subnet for EMR nodes, from the VPC selected above

CoreNodeCount 3 — Number of core nodes to provision (1-20)

InstanceType m4.xlarge — EMR node ec2 instance type - you can add more types by expanding on this list.

OwnerTag — Your name - used to tag the cluster

PurposeTag SparkR Deployment — Purpose - used to tag the cluster

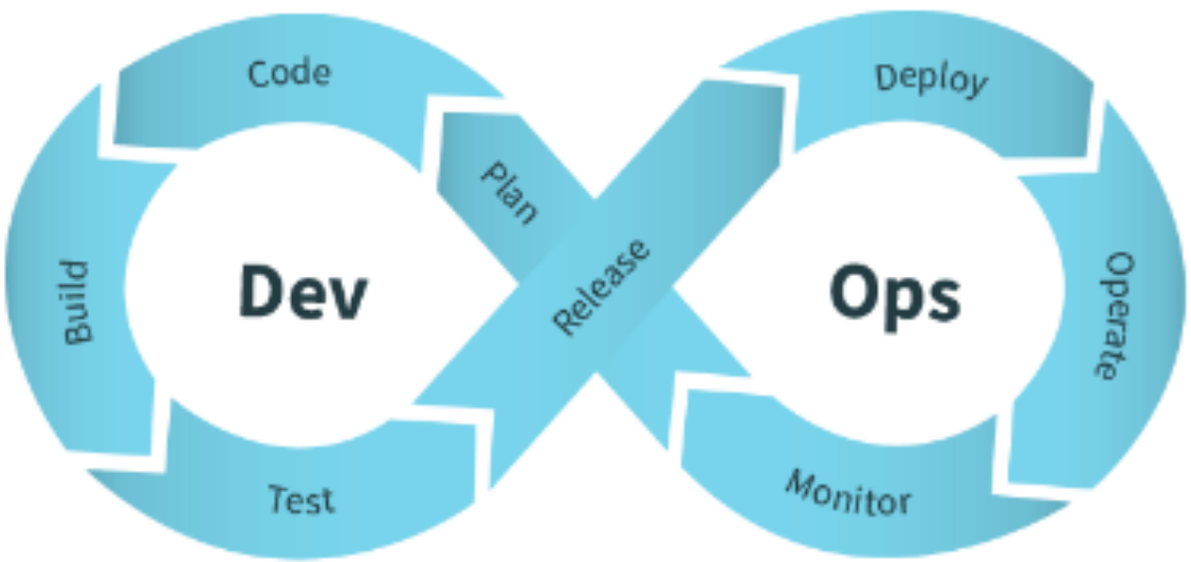
GangliaPort 80 — Ganglia Port

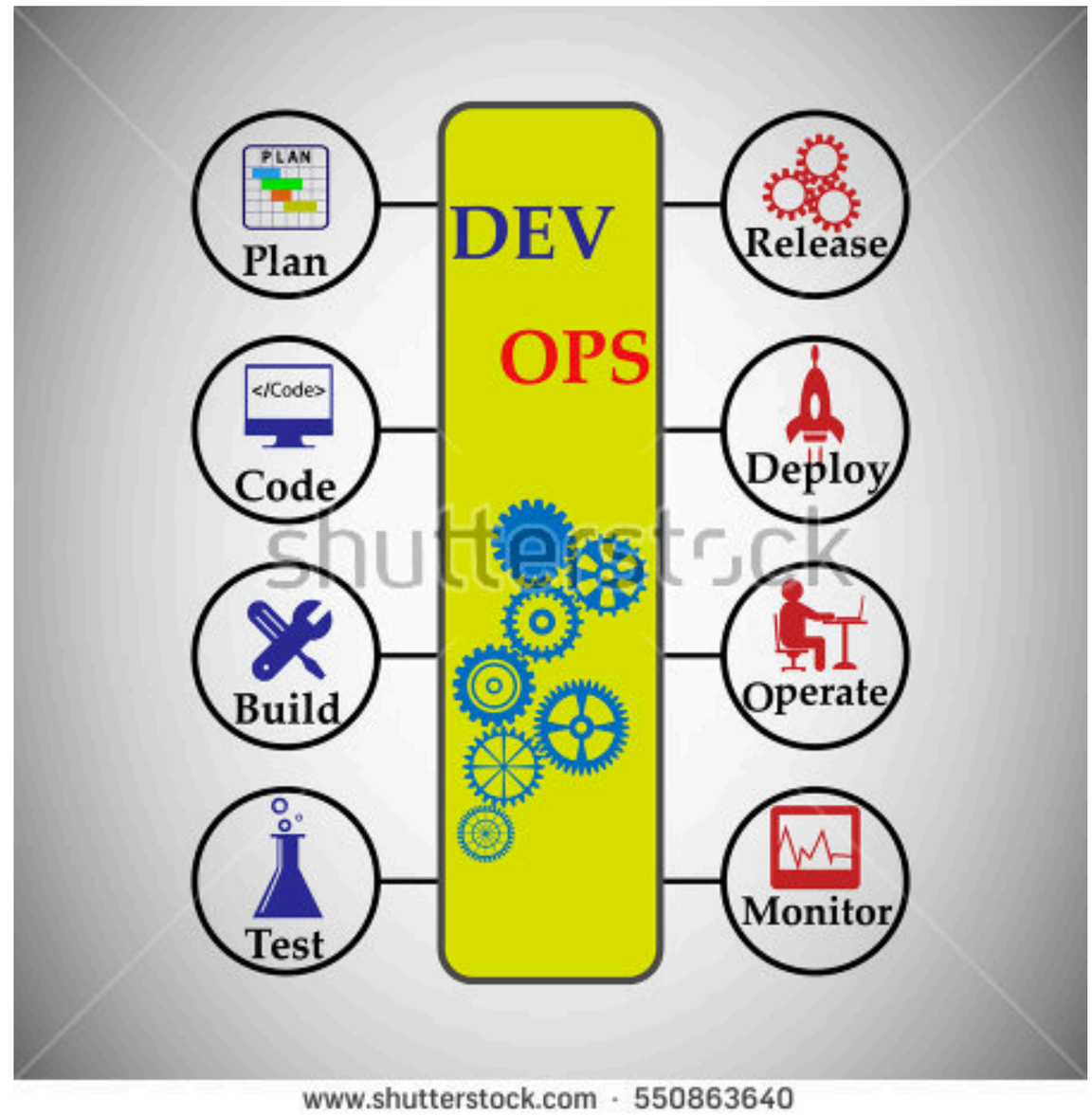
RStudio Options

RStudioPort 4747 — R-Studio Port.

RStudioPassword •••••• — R-Studio Password

Development & Operations





DevOps Principles

- Look for patterns - processes that done repeatedly - and seek to simplify / improve.

- Automate. Frequently, automation will be the mechanism to improve repeated processes.

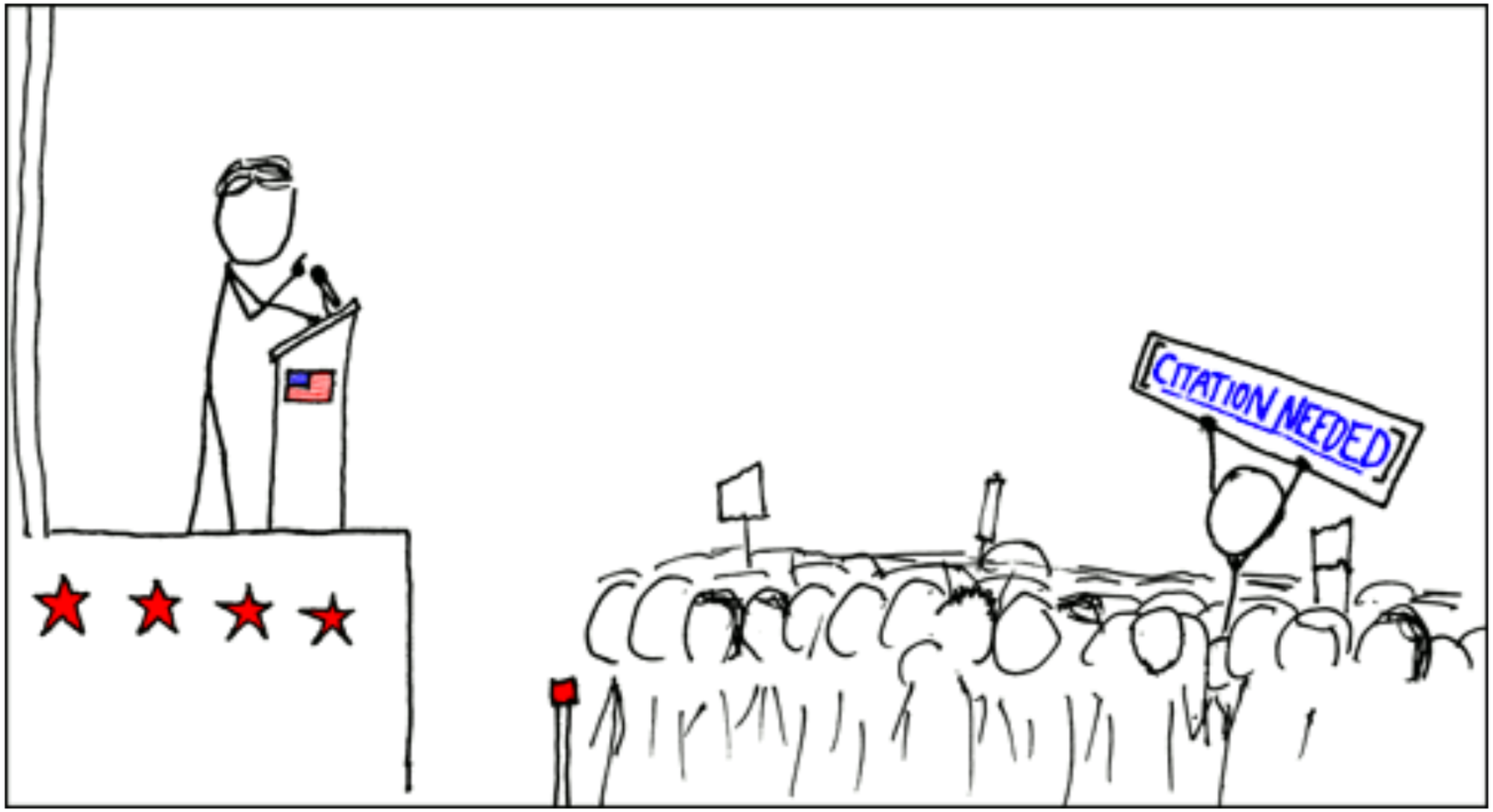
- Continuous Improvement - use streamlined and automated processes to improve your software/tools incrementally.

- Constant Feedback - make getting feedback as easy as possible, allowing you to identify problems in your continuous improvement.

- Fault tolerance & resiliency/redundancy

DevOps & Project Management

- Agile
- Lean
- Waterfall
- Kaban



Infrastructure as Code

Also Called “Programmable Infrastructure”

The goal: “Do once, repeat forever”

Or “idempotence” - think a RESTful API

Infrastructure as Code

- Cloud Compute
- AWS Command Line Interface
- Linux CL & Bash Scripts
- Configuration Management Tools (e.g. Ansible, Vagrant, **Puppet**, Chef).
- Later on:
  - AWS AMI - Amazon Machine Images
  - Containerization and Docker

Puppet

- Open source
- Should run well is Linux, Unix, MacOS, Windows
- Relatively easy learning curve (as opposed to Chef, for instance)
- Written in Ruby but accessible through Puppet DSL
- Stable and mature project (as opposed to Ansible)

Cloud Compute

AWS Launches in 2006 with Elastic Cloud Compute (EC2)

AMAZON EC2

amazon web services

Amazon Web 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
- **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** — 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, iOS, and Web Apps on Real Devices in the Cloud
- **Mobile Analytics** — Collect, View and Export App Analytics
- **SNS** — Push Notification Service

Application Services

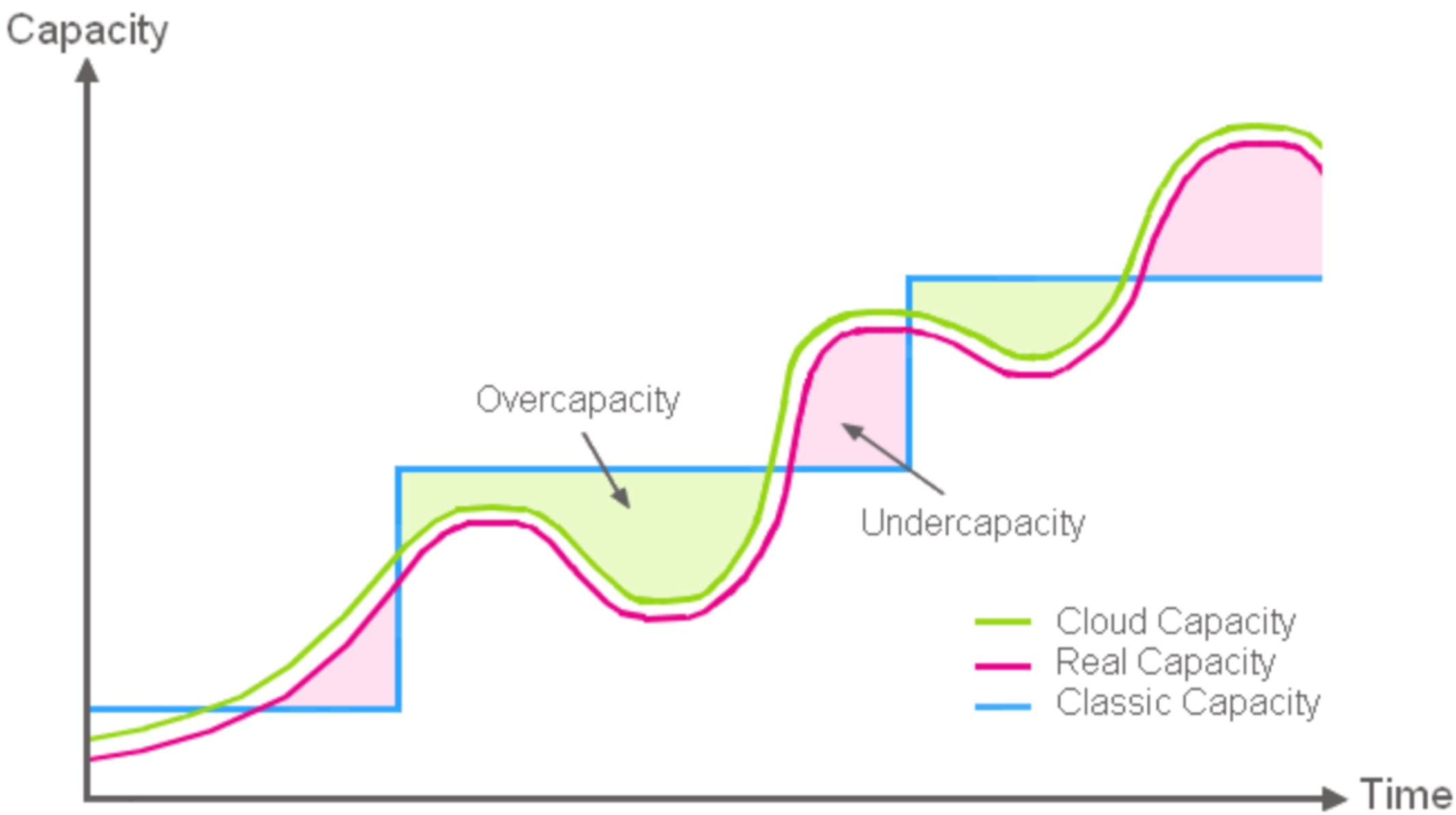
- **API Gateway** — Build, Deploy and Manage APIs
- **AppStream** — Low Latency Application Streaming
- **CloudSearch** — Managed Search Service
- **Elastic Transcoder** — Easy-to-Use Scalable Media Transcoding
- **SES** — Email Sending and Receiving Service
- **SQS** — Message Queue Service
- **SWF**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

Cloud Compute

Elasticity



Cloud Compute

Amazon EMR now supports per-second billing

Posted On: Oct 5, 2017

Amazon EMR is now billed in one-second increments in all AWS Regions. There is a 1 minute minimum charge per instance in your Amazon EMR cluster, and per-second billing is applicable to clusters that are newly launched or already running. The Amazon EC2 instances in your cluster, including On-Demand, Spot, and Reserved instances, and Amazon EBS volumes attached to these instances are billed in per-second increments effective October 2. Pricing is still listed on a per-hour basis, but bills are now calculated down to the second and show times in decimal form. Please visit the Amazon EMR pricing page for more information on per-second billing.

Cloud Compute

	Amazon EC2 Price	Amazon EMR Price
r3.8xlarge	$2.660 per Hour	$0.270 per Hour

Model	vCPU	Mem (GiB)	SSD Storage (GB)
r3.large	2	15.25	1 x 32
r3.xlarge	4	30.5	1 x 80
r3.2xlarge	8	61	1 x 160
r3.4xlarge	16	122	1 x 320
r3.8xlarge	32	244	2 x 320

Cloud Compute

‘Managed’ IaaS (Infrastructure as a Service)

Linux CLI Navigation

- pwd
- ls
- head
- tail - (why is this important in DevOps?)
- cd
  - cd ..
  - cd /

Root Directory

- /bin
- /sbin
- /mnt
- /home
- /root
- /proc

Linux CLI

#!/bin/bash
Or
#!/bin/sh

Set -x -e

Why the e?

Linux CLI Installations

apt-get
curl
yum (use in EMR)

git clone

pip

aws s3 cp

Permissions & Root User

su

or

sudo?

EMR & Instance Information

/mnt/var/lib/info/instance.json

Parameter	Description
isMaster	Indicates that is the master node. Type: Boolean
isRunningNameNode	Indicates that this node is running the Hadoop name node daemon. Type: Boolean
isRunningDataNode	Indicates that this node is running the Hadoop data node daemon. Type: Boolean
isRunningJobTracker	Indicates that this node is running the Hadoop job tracker daemon. Type: Boolean
isRunningTaskTracker	Indicates that this node is running the Hadoop task tracker daemon. Type: Boolean

```
# check for master node
IS_MASTER=false
if grep isMaster /mnt/var/lib/info/instance.json | grep true;
then
  IS_MASTER=true
fi
```

grep - search and return the matching line

-r for searching recursively

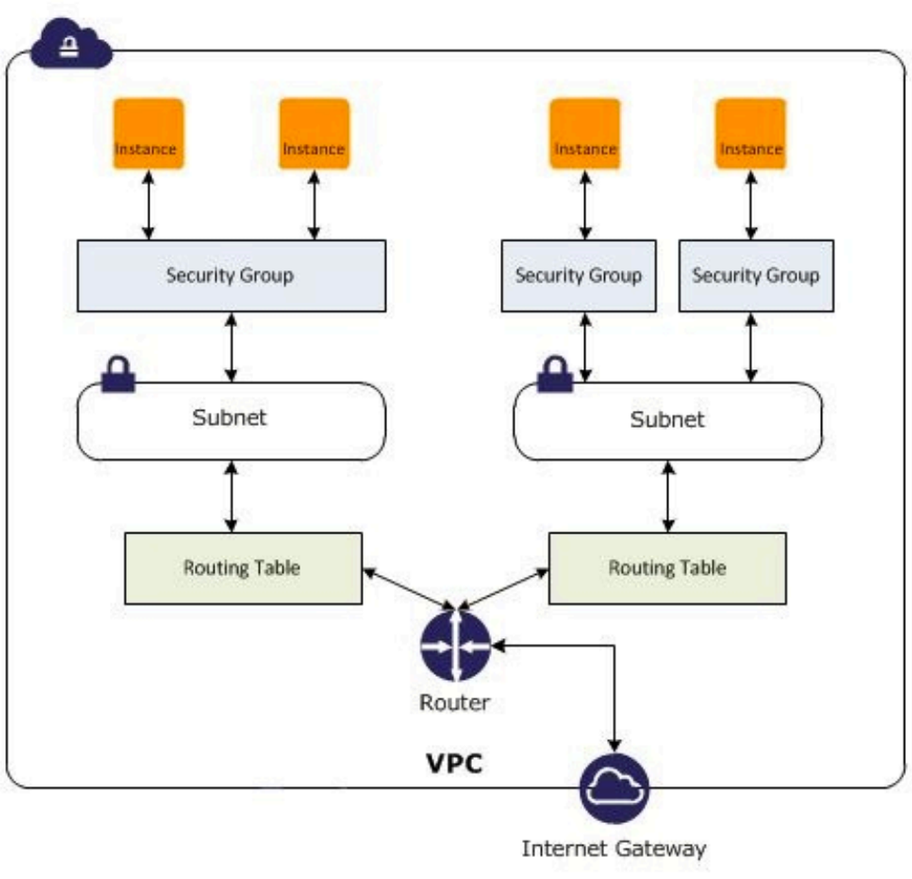
what is the l doing?

```
# get input parameters
while [ $# -gt 0 ]; do
    case "$1" in
    --python-packages)
      shift
      PYTHON_PACKAGES=$1
      ;;
      # do not exit out, just note failure
      error_msg "unrecognized option: $1"
      ;;
    *)
      break;
      ;;
    esac
    shift
done

## User specified python packages go here:
if [ ! "$PYTHON_PACKAGES" = "" ]; then
  sudo python -m pip install -U $PYTHON_PACKAGES || true
fi
```

AWS CLI

```
aws emr create-cluster --release-label emr-5.4.0 ^
  --name 'rstudio-sparkr' ^
  --applications Name=Spark Name=Ganglia ^
  --ec2-attributes KeyName=your-key-pair,InstanceProfile=EMR_EC2_DefaultRole,AdditionalMasterSecurityGroup
  --service-role EMR_DefaultRole ^
  --instance-groups ^
    InstanceGroupType=MASTER,InstanceCount=1,InstanceType=m4.xlarge ^
    InstanceGroupType=CORE,InstanceCount=2,InstanceType=m4.xlarge ^
  --region us-east-1 ^
  --log-uri s3://logs-bucket-name-goes-here ^
  --bootstrap-actions Path="s3://your-bucket-name-goes-here/rstudio_sparkr_emr5lyr-proc.sh"
```



--bootstrap-actions Path="s3://ui-spark-social-science/emr-scripts/rstudio_sparkr_emr5lyr-proc.sh",Args=[--shiny,--no-tutorials] ^

Ganglia!

- Add data to S3
- Create Security Groups
- AWS CLI to Launch a Cluster
  - Include Spark & Ganglia
- Reference a Bootstrap Script for Specific Cluster Installations
  - Jupyter Notebooks (open port)
  - Python and Spark Packages