

# Migrating a Simple Data Analysis Program to the Cloud

*By: Diane Portillo and Paul Lin*

Science Technical Application and Research  
for the Cloud  
(STAR Cloud)

# Agenda

- Background
- Project Goals
- The Cloud
- Area-Averaged Time Series
- Setting-Up the Cloud
- Conclusion

# Background



Paul Lin  
University of Pennsylvania , 2021  
Intended Major: **Earth Science**



Diane Portillo  
DePaul University, 2018  
Major: **Environmental Science**

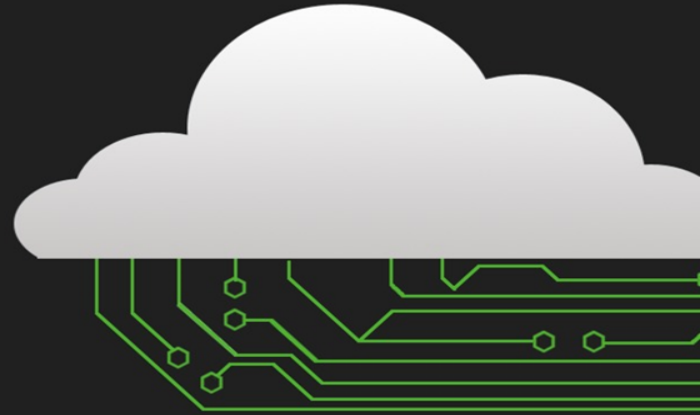
# Project Goals

- Consider and test the advantages of the cloud while documenting and overcoming the pitfalls (so you won't have to!)
- Investigate the cloud's ability to outperform local machines for running Earth science code

# Big Data is Getting Bigger



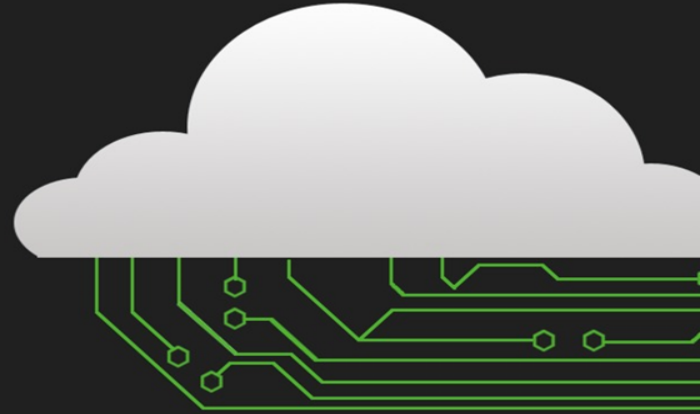
# The Cloud



# What is the Cloud?

*“Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources through a cloud services platform via the internet with pay-as-you-go pricing.”*

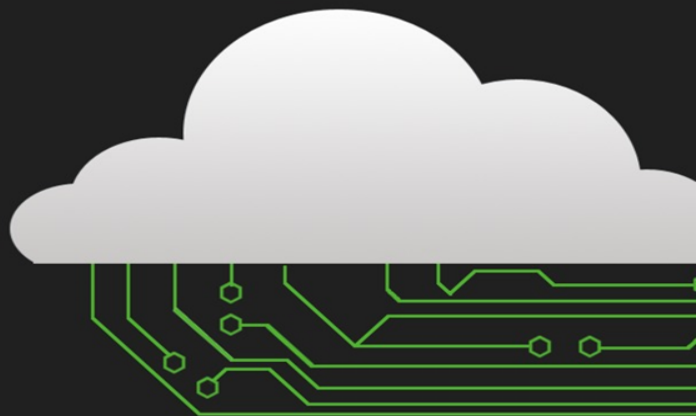
- Amazon Web Services



# Why Cloud?

## Advantages

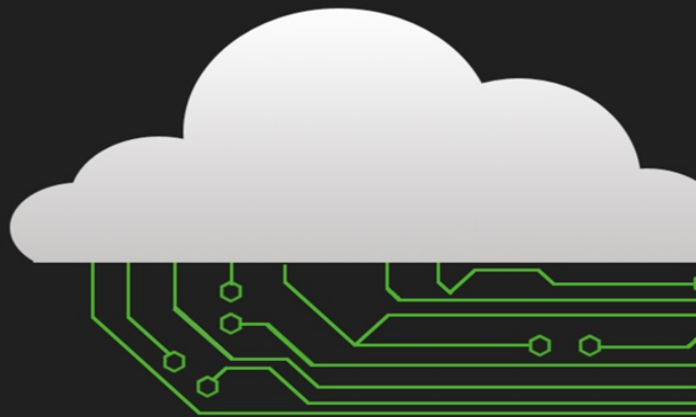
- *Performance:*  
accelerated & pooled computing
- *Ease-of-Use:*  
shared resources for networks,  
storages, servers, and applications
- *Portability:*  
movable datasets
- *Cost-Effectiveness:*  
pay-as-you-go costs to bypass  
excessive upfront costs
- *Elasticity On-Demand:*  
ability to instantaneously scale-up or  
down resources





# For an Earth Scientist?

- **Faster**
  - Commercial cloud CPUs are usually faster than ours...
- **Bigger**
  - Many levels of storage
- **Cheaper**
  - Pay only for what you use
    - CPU
    - Storage



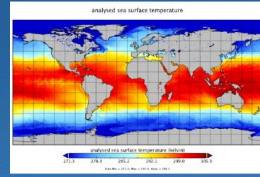
A world map showing a global distribution of values, likely temperature or sea surface temperature, represented by a color scale from dark blue (cold) to red (warm). The map shows a clear latitudinal gradient, with the warmest regions (red) located near the equator and the coldest regions (dark blue) at the poles. The text "Area-Averaged Time Series" is overlaid in white on the map.

# Area-Averaged Time Series

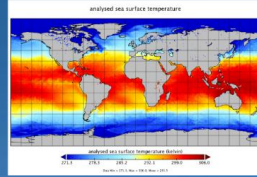
# Area-Averaged Time Series

- Used daily average Sea Surface Temperature (SST) data of 10 days to create a graph
- Parallelize executions

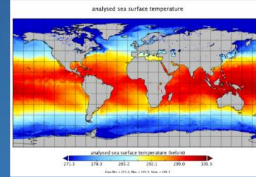
# Area-Averaged Time Series



File A: January 1, 2018



File B: January 2, 2018

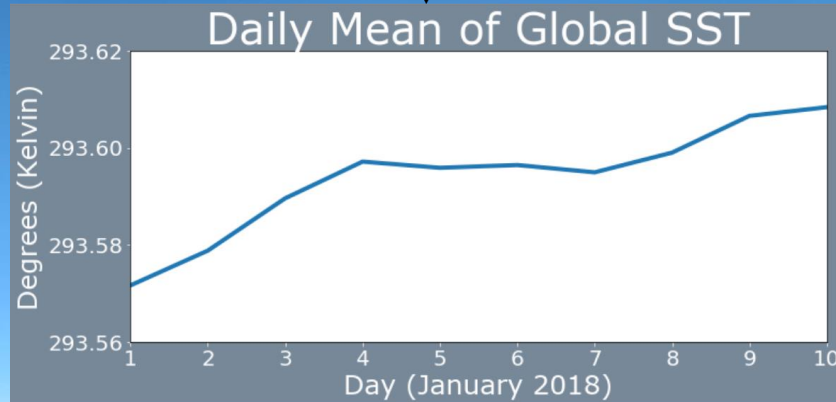


File C-J: Jan 3-10, 2018

Global Mean:  
293.572

Global Mean:  
293.579

Global Mean:  
293.608



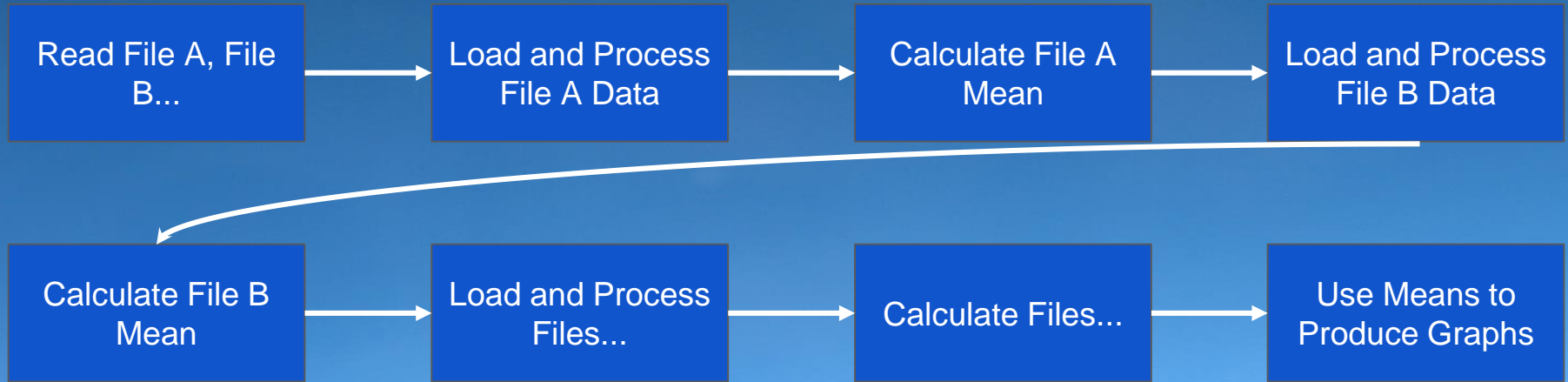
# Area-Averaged Time Series Data

- Data was collected from *PODAAC*, captured by multiple satellite instruments
  - NASA's AMSRE, MODIS on the NASA Aqua and Terra platforms, the US Navy microwave WindSat radiometer, AVHRR on several NOAA satellites, and in situ SST observations from the NOAA iQuam project.
- GHR SST Level 4 MUR Global Foundation Sea Surface Temperature Analysis (v4.1)

# Area-Averaged Time Series Steps

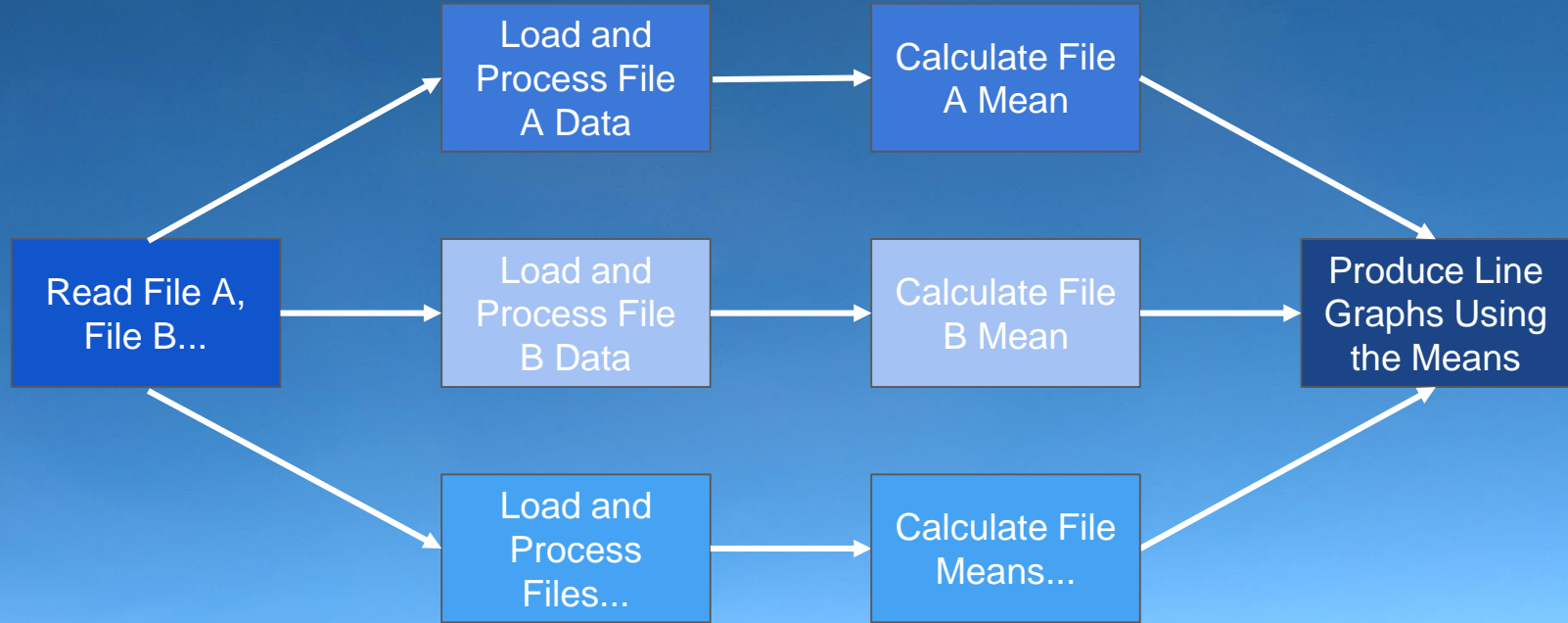
- |                 |   |  |
|-----------------|---|--|
| Read Data Files | I | 1. Import python libraries   |
|                 |   | 2. Read datasets (GHR SST: Group for High Resolution Sea Surface Temperature from MODIS) |
| Process Data    | I | 3. Mask dataset array to account for null values   |
|                 |   | 4. Weigh by latitude array   |
| Calculate Mean  | I | 5. Calculate mean  |
|                 |   | 6. Apply scale factors and additional offsets  |
| Generate Graph  | I | 7. Generate graph based with dates on x-axis and means on y-axis                         |

# Execution Path 1: Serial via “For Loop”





# Execution Path 2: Parallel via “Dask”



Legend:

Process 1

Process 2

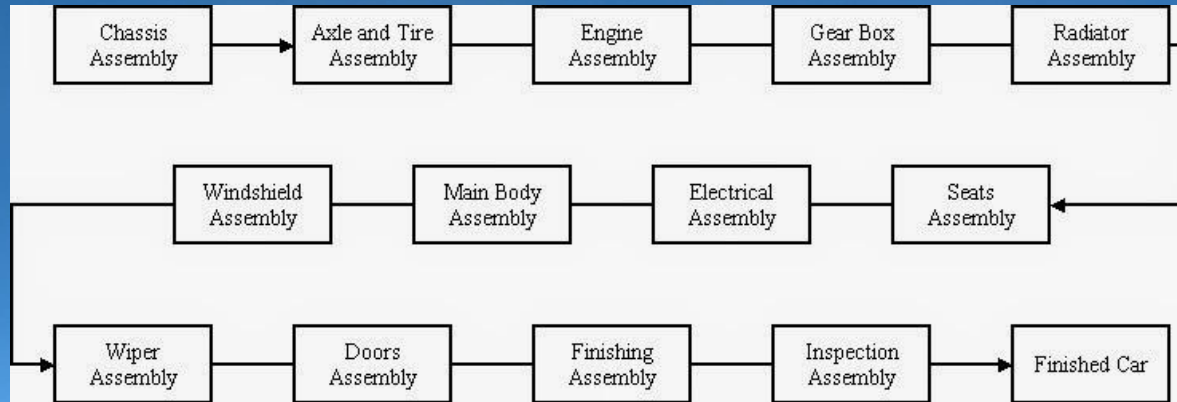
Process 3

Process 4



# What is Dask?

Imagine yourself as a car manufacturing manager...



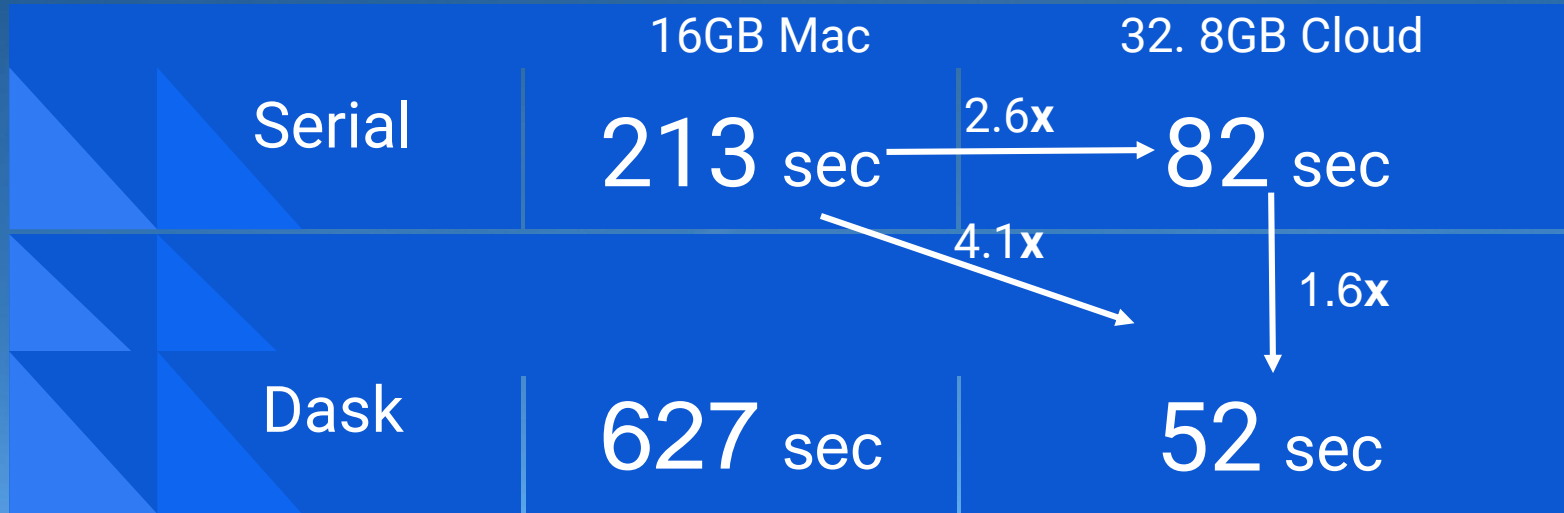
# What is Dask?

Created to scale computational libraries and the surrounding ecosystem of packages

- Parallelization
- Fast simultaneous processing
- Dask hides overhead

```
70 @dask.delayed
71 def mean(weighted_mask):
72     data_mean = weighted_mask.mean().compute(scheduler='threads')
73     return data_mean
74
```

# Execution Runtimes



# Setting-Up the Cloud



## 1. Create an AWS Account 🙄

- Unfortunately, we cannot currently provide guidance here... yet.
- Efforts are underway to provide access to cloud computing to scientists

## 2. Choose, Instantiate, & Configure Virtual Machine (VM) Type

1. Choose AMI2. Choose Instance Type3. Configure Instance4. Add Storage5. Add Tags6. Configure Security Group7. Review

Cancel and Exit

Step 1: Choose an Amazon Machine Image (AMI)  
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. You can select an AMI provided by AWS, our user community, or the AWS Marketplace; or you can select one of your own AMIs.

Quick Start


1 to 35 of 35 AMIs

My AMIs

AWS Marketplace

Community AMIs

☐ Free tier only ⓘ

**Amazon Linux**  
Free tier eligible


**Amazon Linux 2 AMI (HVM), SSD Volume Type** - ami-b70554c8

Amazon Linux 2 comes with five years support. It provides Linux kernel 4.14 tuned for optimal performance on Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1, and the latest software packages through extras.

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

64-bit

**Amazon Linux**  
Free tier eligible


**Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type** - ami-cfe4b2b0

The Amazon Linux AMI is an EBS-backed, AWS-supported image. The default image includes AWS command line tools, Python, Ruby, Perl, and Java. The repositories include Docker, PHP, MySQL, PostgreSQL, and other packages.

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

64-bit

**Red Hat**  
Free tier eligible


**Red Hat Enterprise Linux 7.5 (HVM), SSD Volume Type** - ami-6871a115

Red Hat Enterprise Linux version 7.5 (HVM), EBS General Purpose (SSD) Volume Type

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Select

64-bit

**SUSE Linux**  
Free tier eligible


**SUSE Linux Enterprise Server 12 SP3 (HVM), SSD Volume Type** - ami-3c062943

SUSE Linux Enterprise Server 12 Service Pack 3 (HVM), EBS General Purpose (SSD) Volume Type. Public Cloud, Advanced Systems Management, Web and Scripting, and Legacy modules enabled.

Select

64-bit

## 2. Choose, Instantiate, & Configure Virtual Machine (VM) Type

- VM - template containing software configuration (i.e. operating system, application server, and applications) required to launch your instance
  - Consider the VM size to optimize the necessary storage and memory space
  - *Caveat:* tradeoff exists between size/speed of machine and cost
- 

## 2. Our VM Specs

- *Family*: Storage Optimized
- *Type*: **d2.xlarge**
  - (14 ECUs, 4 vCPUs, 2.4 GHz, Intel Xeon E52676v3, 30.5 GiB memory, 3 x 2048 GiB Storage Capacity)



### 3. Install Python and other necessary python libraries in virtual machine

- Anaconda
  - Contains popular python packages (numPy, Pandas, etc.)
  - Useful for data science
  - Makes for easy deployment for a virtual environment
  - <https://anaconda.org/>
- Other python libraries
  - Dask: enables efficient parallel computations
  - NCO: a command line tool for processing netCDF data

## 4. Getting Datafiles into the Instance

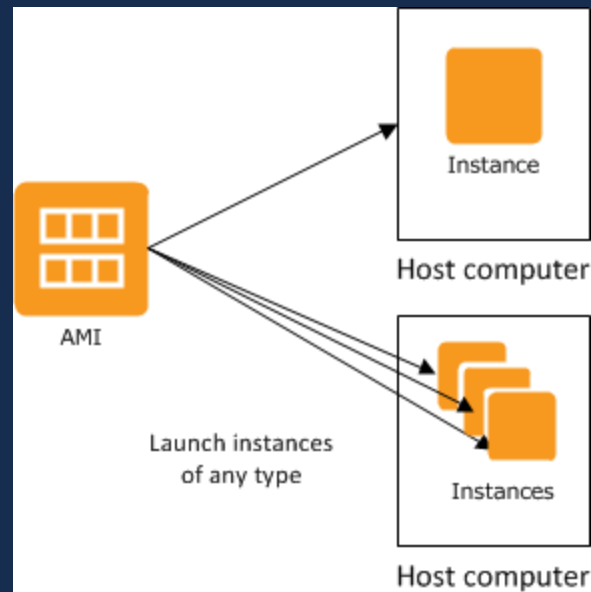
- Use SCP (secure copy protocol)
  - Make sure to be outside of virtual machine
  - `$ scp /local/directory/file.txt  
username@VM_host:destdir`

OR...

- Use wget
    - Downloads files from a network
    - `wget http://website.com/files/file.zip`
- 

## 5. Saving an AMI (Amazon Machine Image)

- A template containing a software configuration (eg. operating system or applications)
- You launch an *instance* (VM), which is a copy of the AMI running as a virtual server in the cloud
- Can launch multiple instances of the same AMI
- Provided by AWS, the community, or create your own
- Can change configurations



# Further Cloud Steps

## 1. Run programs in virtual machine!

(Optional): Sharing Virtual Machines

(Optional): Mounting Elastic Block Storage (EBS) Volume

- *Caveat:* EBS must be in same region as VM, EBS must be dismounted to avoid complications, EBS costs a lot of money!

# Findings

- Setting up cloud VM's can be done by people without programming experience
- Running identical programs yields much faster runtimes in the cloud than in local machines
- Cloud machines required more memory to run programs than local computers, but memory size is elastic
- Increasing the data volume better demonstrates Dask's parallelization advantages

# Possible Future Work

- Experiment with further parallelization methods beyond time (i.e.: geographical areas)
- Incorporate Dask Distributed code to spread computational load across multiple VMs
- Refactor more advanced science algorithms

# Final Words...

You don't need to be a comp. sci person to run analysis faster on cloud because:

- Access to big machines
- Access to packages like dask that parallelize with not a lot of effort

# Further Details

The background of the slide is a dark blue night sky filled with numerous small, white stars. At the bottom of the image, there is a dark, silhouetted horizon line that appears to be a dense forest or a row of trees. The overall color palette is a range of blues, from deep navy to a lighter, dusty blue.



## 4. Access Bastion Host

- SSH into the bastion host
  - Why?
    - Set-up a security group that's configured to listen only on the SSH port (TCP/22)
- Configure (Linux) instances in your VPC to accept SSH connections only from bastion instances.

