

15-319 / 15-619 Cloud Computing

Recitation 2

September 08, 2020

Accessing the Course

- Open Learning Initiative (OLI) Course
 - Access via canvas.cmu.edu
- <http://theproject.zone> (access through canvas)
 - choose CMU as the identity provider
 - AWS Account Setup (AWS, Azure, GCP)
 - Update your [TPZ profile](#) with AWS, Azure & GCP info
 - Complete the Primers on AWS, Azure and GCP
- [Piazza](#)

Amazon Web Services (AWS) Account

- **ONLY IF YOU HAVEN'T DONE SO ALREADY**
- Log on to <https://theproject.zone> through Canvas and make sure you follow the instructions in the Account Setup Primer
- Wait to receive Consolidated Billing Request email from Amazon
 - The linking email is sent automatically, waiting time varies
 - You need to **manually** accept the linking request
- When you receive the linking email, click the link to verify the linked billing
 - Many students have not clicked on the link yet!
 - Check your **SPAM** folder
 - You won't be able to complete the projects without a linked account.

Azure Account

- ONLY IF YOU HAVEN'T DONE SO ALREADY
- Do **not** use your @andrew.cmu.edu or other CMU issued email address.
- Update your TPZ Profile

Google Cloud Platform (GCP) Account

- ONLY IF YOU HAVEN'T DONE SO ALREADY
- Follow the instructions in the primer.
- Receive a \$50 coupon on <https://theproject.zone>
- Redeem the coupon as per instructions on <https://theproject.zone>

Piazza

- Suggestions for using Piazza
 - Discussion forum for a learning community
 - Contribute good questions and answers
 - Read the [Piazza Post Guidelines](#) before asking
- When you have a (project-specific) problem, follow the order below!
 - Try to solve the problem by yourself (Search, Stack Overflow)
 - Read Piazza questions & answers carefully to avoid duplicates
 - Visit TA OHs: TA office hours are posted on Piazza and [Google calendar](#)
 - Create a piazza post
- Please note:
 - **Show the effort you have done first**
 - **Practice how to communicate effectively in a technical setting**
 - **The key to effective communication is to provide the full context. (AndrewId, AWS ID, error message in text, etc.)**
 - Try to ask a public question if possible
 - Don't ask a public question about a quiz question

Provide the context

- Which project module are you working on?
- Which task/section are you working on?
- If relevant, please provide the information of the cloud account and resources.
- Example error message in the plain text format (if you are reporting programmatic issues) or screenshot (if you are reporting UI/UX issues)
 - Please provide example error messages **in the plain text format**, **NEVER** share code/text as screenshots which are not parsable!
 - Use screenshots only for UI/UX issues
- How to reproduce?
- Expected behavior v.s. actual behavior
- Environment summary
- What you have tried?

Articulate technical questions

- There are common patterns to communicate effectively in a technical setting.
- Our course not only aims at building your technical skills, but also training your communication skills.
- We created [a template](#) for you to structure your questions.

Reflecting on Last Week

- AWS, Azure and GCP accounts
- Cloud resources
 - AWS EC2, S3, CloudWatch
 - Azure Compute, Azure Storage
 - GCP Compute, GCP Storage
- Interface
 - Web console, CLI, SDK
- In P0, run a web server, test to access the server over a browser
 - Launch, connect to and terminate VMs
 - Install & run software on a VM
 - Vertical scaling
- Basic SSH skills
- Jupyter Notebook primer
- Infrastructure as Code (Terraform) primer
 - Read it if you have not done so
 - Required by many projects including P1.1

Skill Building in This Course

- Important skill to develop
 - willingness and courage to
 - recognize, explore, experiment and solve problems
on your own with suitable tools
 - ask technical question properly instead of simply asking for solutions
 - learn the basics of new tools quickly and make use of them in a limited time (e.g., 1 week)

Make Sure to Complete the Primers!

- Complete the Primers
 - Understanding AWS/Azure/GCP
 - provisioning resources, connecting to VMs, playing around, ...
 - Linux warmup
 - Git
 - Maven and Checkstyle
 - Jupyter Notebook
 - Command Line
 - Data analysis in Bash
 - Data analysis in Python (pandas)
 - Infrastructure as Code (Terraform)

Programming Experience Expected

- **Strong proficiency** in at least one of the following, with some fair comprehension of the others:
 - **Java 8**
 - **Python 3**
 - **Bash**
- **Java and Python are required** to complete parts of Projects.
- Use the time now to brush up
- Please read Maven primer!
- Do not fear bash/python scripting, it will make your life easier!

Completing Projects in this Course

- Provision AWS, Azure or GCP Resources
 - Use the Cloud VM Images we provide for the project
 - Tag all instances!
- Monitor your cost
 - Calculate costs before you provision!
- Complete tasks for each project module
 - Each project module has several sections unlocked by AssessMe
- Submit your work
 - Pledge of integrity
 - Results in scoreboard
- Terminate all resources when you have verified your score and kept a copy of your work (e.g., git **private** repo)

Tagging

- Tag **all** tag-able resources on AWS
 - Before you make a resource request, read the docs/specifications to find out if tagging is supported
 - We will specify which resources are required to be tagged in each project
 - Apply the tags during resource provisioning
 - We need tags to track usage, a grade penalty will be applied automatically if you do not tag!
 - Spot instances
 - Tags of a spot request do NOT propagate to the VMs!
 - AWS EC2 Fleet is the remedy
- Tagging Format
 - Key: Project
 - Value: 0, 1.1, 1.2....etc.

Budgets and Penalties

- No proper tags → 10% grade penalty
- Provision resources in regions other than us-east-1 → 10% grade penalty
- Budget
 - For P1.1, each student's budget is **\$1**
 - Exceeding Budget → 10% project penalty
 - Exceeding Budget x 2 → 100% project penalty (no score)
 - You can see Cost and Penalties in TPZ
- No exceptions
- We give you an opportunity to learn in Project 0 without affecting your grade
- We will enforce these penalties automatically starting from Project 1.1

How to Work on a Budget

- P1.1 Budget → \$1
- You are only allowed to use t3.micro
 - \$0.0104 per hour (on demand)
- Other costs to consider:
 - General Purpose SSD (gp2) EBS Volumes is \$0.1 per GB/month
 - Instances using our AMI gets 8 GB EBS by default.
 - Data transfer costs (minimal)

Total time:

$$\text{\$1} / (\text{\$0.0104} + 8 * \text{\$0.1} / 30 / 24) = 86.87 \text{ hours}$$

- **Note:** Free Tier does not apply to linked accounts!

Academic Integrity Violation

- Cheating → the lowest penalty is a 200% penalty & potential dismissal
 - Other students, previous students, Internet (e.g. Stackoverflow)
 - Do not work on code together
 - This is about you struggling with something and learning
 - Penalty for cheating is SEVERE – don't do it!
 - Ask us if you are unsure

Compromised Accounts

- If you put any of your credentials in files on
 - Github, Dropbox, Google Drive, Box, etc.
 - You are vulnerable to getting your account compromised.
 - Going over 2x the project budget \Rightarrow 100% penalty!
- People are scanning publicly available files for cloud credentials.
 - They compromise your account and launch resources in other regions.

DO NOT SAVE YOUR CLOUD CREDENTIALS IN FILES!

Deadlines!

- **Hard Deadlines**
 - No late days, no extensions
 - Start early!
 - Plan your activities, interviews and other commitments around the deadlines.
 - **No exceptions!**
- Project modules are due on Sundays at 23:59 ET
- Quizzes are typically due on Fridays
 - There is one exception this semester before Spring Break: Quiz 7 is on Thursday, October 22

Deadlines!

- **Project deadlines**
 - **On TheProject.Zone**

- **Quiz deadlines**
 - **On OLI**

Quiz 1 Preparation

- Tests your understanding in Modules 1 and 2
 - Cloud computing fundamentals, service models, economics, SLAs, security
 - Use the activities in each page for practice.
 - You will be tested on you ability to perform the stated learning objectives on OLI:

Module 1 / Cloud Computing Overview

1

LEARNING OBJECTIVES

Explain the concept of cloud computing	Briefly understand how computing systems across domains dealt with scale before the cloud	Briefly recall the recent history of cloud computing, illustrating its evolution
List some of the enabling technologies in cloud computing, and discuss their significance	Differentiate cloud service models, such as IaaS, PaaS, and SaaS	Enumerate the different types of clouds, and compare and contrast them
List some of the common cloud providers and their associated cloud stacks	Recall popular cloud use case scenarios	

Module 2 / Economics, Benefits, Risks, Challenges and Solutions

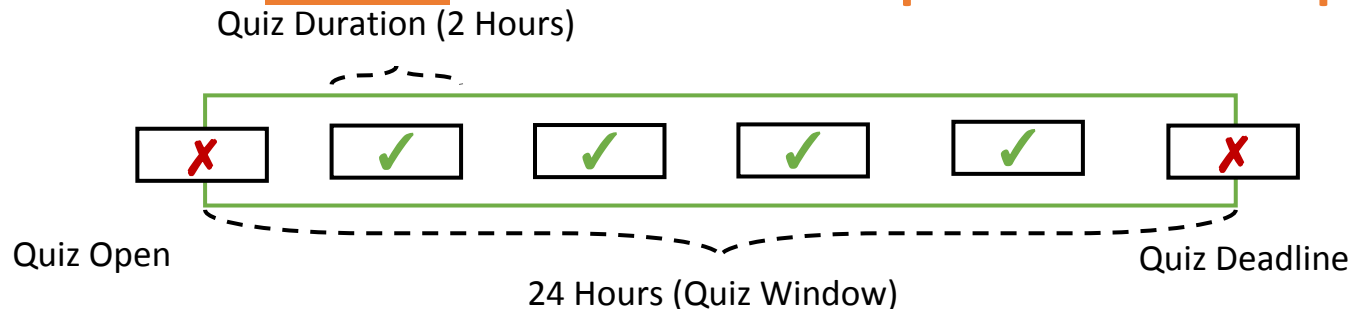
11

LEARNING OBJECTIVES

Discuss some of the advantages and disadvantages of the cloud paradigm	Articulate the economic benefits as well as the issues/risks of the cloud paradigm for users	Articulate the economic benefits as well as the issues/risks of the cloud paradigm for cloud service providers
Define SLAs and SLOs and illustrate their importance in Cloud Computing	Enumerate and explain various threats in cloud security	Enumerate and explain various controls in cloud security

Quiz 1 Logistics

- Quiz 1 will be open for 24 hours, Friday, Sept 11
 - All quizzes are **open-book** tests.
 - Quiz 1 becomes available on **Sept 11, 00:01 AM ET.**
 - Deadline for submission is **Sept 11, 11:59 PM ET.**
 - Once open, you have **120 min** to complete the quiz.
 - You may not start the quiz after the deadline has passed.
 - Every 15 minutes you will be prompted to save.
 - **Maintain your own timer from when you start the quiz.**
 - **Click submit before deadline passes. No Exceptions!**



Submit Before Deadline

- When you start the Quiz, you cannot stop the clock.
 - You have 120 minutes to click on submit.
 - You have to keep track of the time yourself.
 - If you don't click on submit you will not receive a grade.

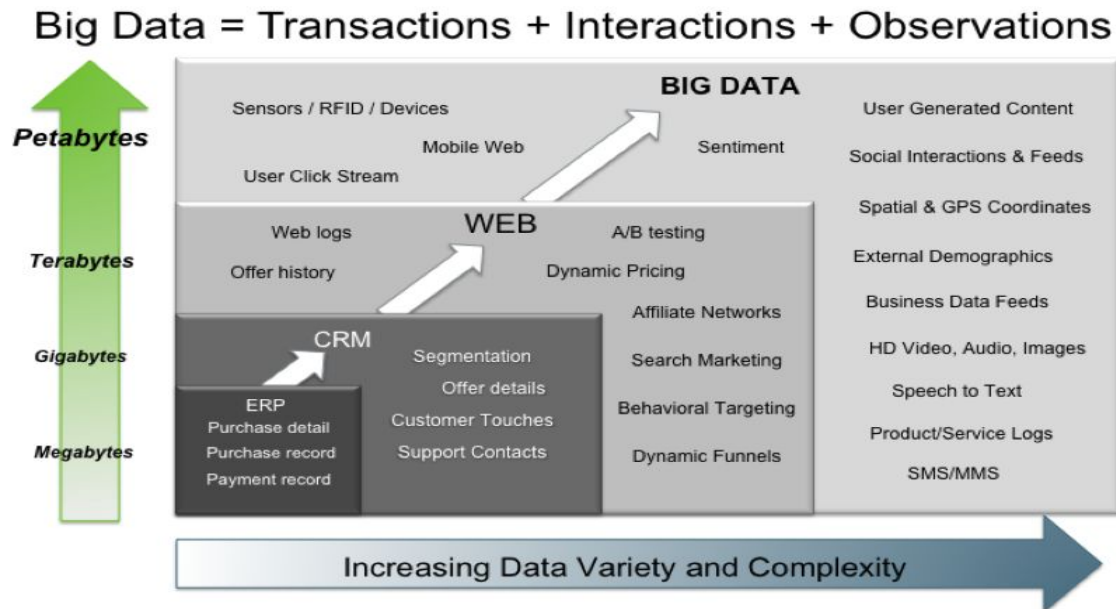
**YOU MUST SUBMIT
WITHIN 120 MINUTES
AND
BEFORE THE DEADLINE**

Do not collaborate on quizzes

- In previous semesters, there is always a significant minority who decided to collaborate on quizzes, especially at the semester start and when the team project began.
- **We have to emphasize again that unauthorized collaboration on quizzes is also AIV.**

Project 1 Motivation: Big Data

- What is Big Data?
 - It is high volume, high velocity, and/or high variety information assets.
 - There is a lot of value in the analysis of big data for organizations



Source: Contents of above graphic created in partnership with Teradata, Inc.

Use Cases: Big Data Analysis

- Online retailers are analyzing consumer spending habits to learn trends and offer personalized recommendations and offers to individual customers.
- Companies such as Time Warner, Comcast etc. are using big data to track media consumption habits of their subscribers and trends to provide value-added information to advertisers and customers.

Trending Topics are Everywhere!

TRENDING

- **AMBER Alert:** US Facebook users to receive alerts within news feed regarding local missing children
- **Dazed and Confused:** Actor Matthew McConaughey's audition tape for 1993 comedy movie surfaces
- **Penny Dreadful:** Showtime premieres full trailer for season 2 of horror series
- **Mega Millions:** Retired school principal wins \$326 million, biggest prize in New York Lottery history
- **Ted Cruz:** Senator, R-Texas, named chair of Subcommittee on Space, Science and Competitiveness
- **#KissAGingerDay:** Social media campaign promotes displaying affection toward redheads
- **Tim Willcox:** BBC reporter apologizes to daughter of Holocaust survivor for Israel, Palestine comment
- **Ashley Furniture Industries:** Chain write off \$1.5 million in customer purchases for Ohio State win
- **#PrayForNigeria:** Hashtag campaign takes off on social media in response to Boko Haram attack in Baga
- **David Cameron:** WhatsApp, Snapchat could be banned in Britain with prime minister's surveillance plan

Trends · Back to Discover

- #EveryVillainNeeds 🔥 Promoted
- #ElderlyTVShows
- #OscarNoms
- #Empire
- Lego Movie
- Unfriended
- Pittsburgh
- #KiranBedi
- Xbox One
- #MyLastWordsIn5Words

Trending Now

- Oscar Nominations**
500,000+ searches
Image Source - New York Times (blog)
United States
- Cardale Jones**
100,000+ searches
Image Source - USA TODAY
- Martin Luther King Jr**
100,000+ searches
Image Source - Columbus Dispatch

Why Trending Topics?

- Identify trends and viral content
- Maximize advertisement placement opportunities
- Search Engine Optimization (SEO)
- And more....

Project 1

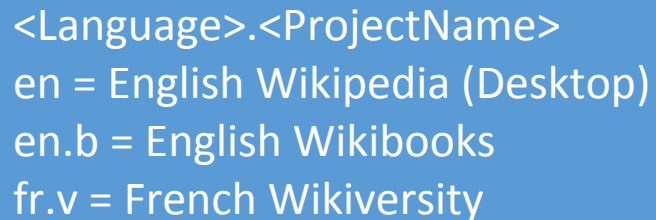
- Identify Trending Topics on Wikipedia
 - Use the hourly pageviews dataset.
- Project 1.1: (This Week)
 - Find trends from a single hour of data.
- Project 1.2: (Next Week)
 - Find trends with the 30-day dataset using MapReduce.
 - Data from March 8 to April 6 in 2018

The Dataset

- Data set
 - [Wikimedia page views dataset](#)
 - One File Per Hour

- Format:

`<domain code> <page title> <number of accesses> <total data returned>`



`<Language>.<ProjectName>`
en = English Wikipedia (Desktop)
en.b = English Wikibooks
fr.v = French Wikiversity

Project 1.1 Tasks

- Task 1: Sequential data pre-processing
 - Implement sequential data filter in Java
 - Practice test-driven development (TDD) with JUnit
 - Achieve 100% code coverage for the DataFilter class
- Task 2: Data analysis
 - Search a file with grep
 - Filter or process data with awk
 - Data Analysis with Jupyter Notebook and Pandas library
 - Identify the limitations of the sequential programming

Data Pre-processing is Important

- Impossible: Raw Dataset \rightarrow Data analysis
- Raw Dataset \rightarrow Data pre-processing \rightarrow Data analysis

raw data:



after data pre-processing



Task 1: Data Pre-processing

- We are only interested in English Wikipedia desktop/mobile pages (**<domain code>**: en, en.m)
- This dataset is raw, real-world
 - Never assume that the dataset is perfectly clean and well formed
- Use the filtering rules specified in the writeup
- If there are records from both desktop and mobile sites for the same page title, sum the accesses into one record
- Sort the pages by number of pageviews, break ties by ascending lexicographical order
- Output: **<page title> <number of accesses>**

Bad Coding Practices!

```
public static void main(final String[] args) {  
    read the records from the input  
    for record in records:  
        if it violates the rule A: (20 lines)  
            continue  
        if it violates the rule B: (20 lines)  
            continue  
        ... 5 other rules (100 lines)  
        put record into a map <title, pageview>  
    sort the map  
    print output  
}
```

Good Coding Practice: Test-Driven Development (TDD)

What is TDD?

- divide the problem into a series of small steps
- start by writing test cases
- then refactor the code to pass the test
- and repeat
 - Test case \Rightarrow code \Rightarrow pass \Rightarrow another test case \Rightarrow ...

TDD emphasizes writing unit tests ahead of writing the code.

TDD lets you treat failures as a norm instead of an exception.

Test-Driven Development (TDD)

Why TDD?

- Helps to structure your code in a way that easily facilitates testing
- Separates the concerns and makes your code clean, easy-to-read and robust
- Ensures that your changes won't break existing functionality
- Achieves safer refactoring, increasing returns and effective collaborations
- TDD is an industry best practice!!!

TDD w/ JUnit 5: Start by signature

```
public class Filter {  
  
    public static boolean containsCloud(final String record) {  
        // it is okay to start with an incorrect solution  
        // the method signature is what matters  
        return false;  
    }  
}
```

TDD w/ JUnit 5: Create test cases

```
import org.junit.jupiter.api.Test;
```

```
import static org.junit.jupiter.api.Assertions.*;
```

```
class FilterTest {
```

```
    @Test
```

```
    void testContainsCloud() {
```

```
        // positive
```

```
        assertTrue(Filter.containsCloud("cloud computing"));
```

```
        // negative
```

```
        assertFalse(Filter.containsCloud("mapreduce"));
```

```
    }
```

```
}
```

TDD w/ JUnit 5: Run the test

```
[INFO] Running FilterTest
```

```
[ERROR] Tests run: 1, Failures: 1, Errors: 0, Skipped: 0, Time elapsed: 0.021 s <<<  
FAILURE! - in FilterTest
```

```
[ERROR] testContainsCloud Time elapsed: 0.017 s <<< FAILURE!
```

```
org.opentest4j.AssertionFailedError: expected: <true> but was: <false>
```

```
    at FilterTest.testContainsCloud(FilterTest.java:9)
```

```
[INFO] Results:
```

```
[ERROR] Failures:
```

```
[ERROR]    FilterTest.testContainsCloud:9 expected: <true> but was: <false>
```

```
[ERROR] Tests run: 1, Failures: 1, Errors: 0, Skipped: 0
```

```
[INFO]
```

```
[INFO] -----
```

```
[INFO] BUILD FAILURE
```

```
[INFO] -----
```

TDD w/ JUnit 5: Implement

```
public class Filter {  
    public static boolean containsCloud(final String record) {  
        return record.contains("cloud");  
    }  
}
```


TDD w/ JUnit 5: Rerun the test

```
[INFO] T E S T S
```

```
[INFO] -----
```

```
[INFO] Running FilterTest
```

```
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.012
```

```
s - in FilterTest
```

```
[INFO]
```

```
[INFO] Results:
```

```
[INFO]
```

```
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
```

```
[INFO]
```

```
[INFO]
```

```
-----
```

```
[INFO] BUILD SUCCESS
```

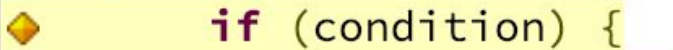


```
[INFO]
```

```
-----
```

Java Code Coverage Tool (JaCoCo)

Jacoco analyzes Java byte code and maps the collected information back to source code to visualize the code coverage at line-level granularity.




















- Instructions (C0 Coverage)
- Branches (C1 Coverage)

```
14.     public static String trueOrFalse(boolean condition) {
15.          if (condition) {
16.              return "true";
17.     } else {
18.              return "false";
19.     }
20. }
```

Your JaCoCo Task

project1 > edu.cmu.scs.cc.project1 > DataFilter

DataFilter

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods
checkSuffix(String[])		0%		0%	3	3	4	4	1	1
checkFirstLetter(String[])		0%		0%	3	3	6	6	1	1
checkAllRules(String[])		0%		0%	7	7	6	6	1	1
static {...}		100%		n/a	0	1	0	3	0	1
checkPrefix(String[])		100%		100%	0	3	0	6	0	1
checkSpecialPage(String[])		100%		100%	0	3	0	4	0	1
lambda\$sortRecords\$0(Map.Entry, Map.Entry)		100%		100%	0	2	0	4	0	1
checkDomain(String[])		100%		100%	0	3	0	1	0	1
sortRecords(TreeMap)		100%		n/a	0	1	0	4	0	1
checkDataLength(String[])		100%		100%	0	2	0	1	0	1
getColumns(String)		100%		n/a	0	1	0	1	0	1
Total	73 of 422	82%	20 of 36	44%	13	29	16	40	3	11

Your task is to create the test cases to achieve 100% instructions and branches coverage.

P1.1 Task 1: Data Pre-processing Code Template

- In this task, we provide a code template:
 - Merges both desktop and mobile sites for the same page title if any
 - Sorts the output in descending numerical order of the number of accesses and break ties by ascending lexicographical order
 - Outputs the results into a file named as exactly “output”
- It also defines a set of filter methods that you need to implement
- We provide a set of test cases for the first several filter methods
- Your task is to:
 - add the test cases for the rest of the required methods
 - achieve 100% code coverage
 - implement the methods and pass the test
 - make the program encoding aware

Develop Robust Code that Can Execute Correctly on Multiple Environments

- “It works on my laptop, why does not it work when it is deployed to production?”
- “My submitted code does not produce the same results as the one on my EC2 instance...”
- If your code behaves well in your development environment, it does **not** guarantee that your code will work perfectly in other environments.
- If you run into this, read the writeup carefully, check and adopt best practices before you create posts on Piazza.
- Be cautious about implicit reliance on your environment
 - Locale
 - Encoding-aware I/O
 - Newline(EOL)
 - Versions & Compatibility
 - Absolute/Relative Paths



Develop Robust Code that Can Execute Correctly on Multiple Environments(cont.)

- Example of how *locale* will change default behavior

	LC_ALL=en_US.UTF-8	LC_ALL=C
sort	aAbB	ABab
encoding	ьмопяёя	????????????????
...

Develop Robust Code that Can Execute Correctly on Multiple Environments(cont.)

- newlines, Windows versus Linux
 - `\n`
 - `\r\n`
- Versions & compatibility
 - `python2` `python3` over `python`
 - `pip2` `pip3` over `pip`
- Absolute versus relative paths
 - use relative ones!

Do I have a remedy to this endless checklist? Is there a solution to guarantee that the software will always run in the same way, regardless of its environment? The answer is yes (Project 2).

Progressively Solve Data Science Problems with Jupyter Notebook

Why Jupyter Notebook?

- Interactive Computing
 - store intermediate data in memory
 - implement and execute your solution cell by cell
- Persisted Output and Reproducible Analysis
 - write data analysis reports and share with others





Your Data Science Task with Jupyter Notebook

- Finish the Jupyter Notebook primer
- Visit the Azure Notebooks library [15-319/15-619: Cloud Computing Course](#)

CloudComputingCourse > Libraries > cloud-computing-course

▶ Run + New ⚙ Settings 🗄 Share 📄 Clone 1 Clone ☆ Star (0) 🖥 Terminal ■ Shutdown 🖨 Preview 📄 Edit File ↓ Download

Search Show hidden items

FILE NAME ▼	FILE TYPE
 DataAnalysisInBash.ipynb	Notebook
 DataAnalysisInPython.ipynb	Notebook
 HeadFirstCommandLine.ipynb	Notebook
 README.md	Markdown

Project 1.1 Workflow

- Launch EC2 instance with a special AMI
 - Experiment using Terraform with prepared worked examples
- Download the required dataset
- Implement the Data Filter program
 - Achieve 100% code coverage
- Complete Data Analysis Task
 - Answer the awk/grep questions inside runner.sh
 - Answer the Python questions inside data_analysis.ipynb
 - **Answer q10 which will be manually graded**
- Submit your code for grading
 - Complete the references file in JSON format
 - Execute `submitter` to submit your code
- Finish Project Reflection (graded) before the deadline

- Finish Project Reflection Feedback for 3 students
 - Within 7 days **after** the project deadline

Grading of Your Projects

- Code submissions are auto-graded
- Scores will be made available on <http://theproject.zone>
 - it may take several minutes for your score to show
 - the submissions table is updated with every submission
- We will grade all the code (both auto and manually graded)
- **Hard to read code** of poor quality will lead to a loss of points during manual grading.
- Lack of **comments**, especially in complicated code, will lead to a loss of points during manual grading.
- Poor **indentation** will lead to a loss of points during manual grading
 - Preface each function with a header that describes what it does
 - Use descriptive variable and function names
 - Use Checkstyle, PEP8, or other tools to check your coding style
- The idea is also **NOT** to comment every line of code

Reminder: Deadlines

- **This Friday** at 23:59 ET
 - Quiz 1
- **This Sunday** at 23:59 ET
 - Project 1.1 (including Project Reflection)
- **Next Sunday** at 23:59 ET
 - Project 1.1 Reflection Feedback
- **ASAP, at the latest 9/14/2020 at 23:59 ET**
 - Academic Integrity Course Quiz