IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

DATA WRANGLING IN THE CLOUD, AWS WORKFLOW FOR BUILD LAYER AWS CLOUD-FORMATION AND SYSTEMATIC STUDY OF DATA WRANGLING USING PYTHON

[1]Anup Kumar, [2]Jitendra Kumar, [3]Shubham Kumar, [4]Mohd Shahnawaz
[1]Assistant Professor, [2] Assistant Professor, [3]Assiatant Professor, [4]Assistant Professor
[1,2,3,4] Dept. of computer science,
[1,2,3,4] Shobhit University Gangoh, Saharanpur, India.

Abstract: Python is an interpretable and scriptable programming language. That is object oriented, the primary data wrangling, cloud computing and machine learning. The philosophy, architecture, and application of the data wrangling process—which is utilized in business intelligence and data warehousing—are presented in this work. The art of transforming or preparing data is known as "data wrangling." It is a technique designed for fundamental data management, where data must be appropriately formed, processed, and made available for the most convenient usage by possible users in the future. To support extensive ad-hoc queries, a lot of historical data is either aggregated or kept in data warehouses as facts or dimensions. Data wrangling makes it possible to process business queries quickly and provide analysts and end users with the appropriate answers. The wrangler suggests predicted transcription scripts and uses interactive language. This facilitates the user's understanding of the elimination of manual iterative processes. The best examples in this case are decision support systems. Big data principles have a significant impact on the methods used to prepare data for mining insights, from self-service analytics and visualization tools to the data source layer. Python is an interpretable and scriptable programming language that can be used for both learning and practical applications. Guido van Rossum created the potent high-level language python. It is an interpretable programming language that is object-oriented. The primary Python programming software tools for data wrangling, cloud computing, and machine learning approaches will be introduced in this presentation. In summary, this paper will begin with an introduction to Python programming and data wrangling.

Index Terms - AWS, Cloud Computing, Data Wrangling, Storage.

1- Introduction to AWS

Amazon Web Services (AWS) is a leading top platform in providing the web services of various domains. AWS follows the trends of digital IT and comes up needy services with optimized performances covering a wide range of services from Compute to Storage. It covers a wider range of customers of different domains to expand their business operations. This Article covers the fundamentals of AWS and its scope of IT business. Amazon Web Services (AWS) is a leading cloud platform that offers a wide range of cloud computing services to help businesses and developers build, deploy, and manage applications and services efficiently. AWS provides essential services like compute power through EC2 (Elastic Compute Cloud), storage solutions with S3 (Simple Storage Service), and managed databases with RDS (Relational Database Service). Additionally, it offers advanced networking capabilities via VPC (Virtual Private Cloud) and content delivery through Cloud Front. Security and compliance are also robust, featuring tools such as IAM (Identity and Access Management) and KMS (Key Management Service) to ensure data protection and secure access management. AWS's flexibility, scalability, and pay-as-you-go pricing model make it an attractive choice for organizations of all sizes, enabling them to innovate rapidly while managing costs effectively.

Getting started with AWS involves creating an account, exploring the AWS Management Console, and utilizing the Free Tier to gain initial experience without incurring costs. AWS's extensive documentation, tutorials, and developer community support users in learning and leveraging the platform's capabilities. Businesses can deploy their first applications quickly, benefiting from AWS's global infrastructure, which ensures high availability and low latency. With AWS, organizations can scale their operations seamlessly to meet varying demand, enhance their security posture, and tap into the latest technological advancements, ultimately driving business growth and operational efficiency.

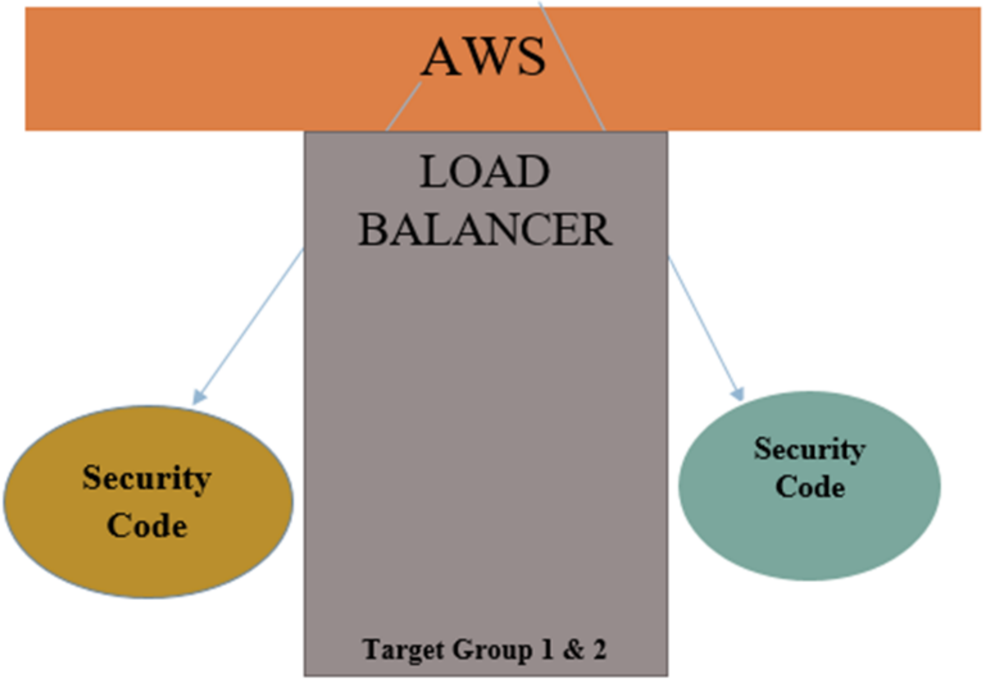

Figure 1. Basic diagram of AWS workflow.

1.1 Introducing data wrangling

For organizations to become data-driven to provide value to customers or make more informed business decisions, they need to collect a lot of data from different data sources such as clickstreams, log data, transactional systems, and flat files and store them in different data stores such as data lakes, databases, and data warehouses as raw data. Once this data is stored in different data stores, it needs to be cleansed, transformed, organized, and joined from different data sources to provide more meaningful information to downstream applications such as machine learning models to provide product recommendations or look for traffic conditions. Alternatively, it can be used by business or data analytics to extract meaningful business information.

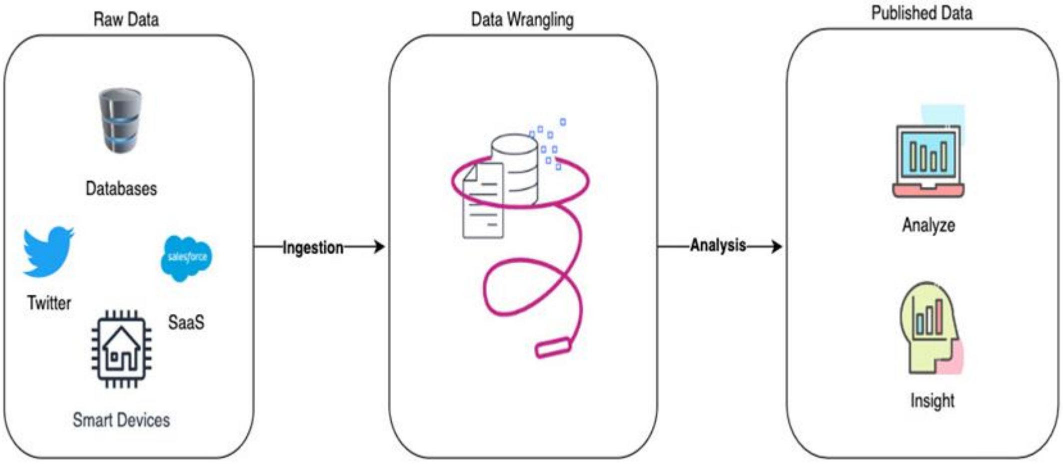

Figure 2. Process of data wrangling.

1.2 Advantages of data wrangling

If we go back to the analogy of oil, when we first extract it, it is in the form of crude oil, which is not of much use. To make it useful, it has to go through a refinery, where the crude oil is put in a distillation unit. In this distillation process, the liquids and vapors are separated into petroleum components called fractions according to their boiling points. Heavy fractions are on the bottom while light fractions are on the top, as seen here in diagram:

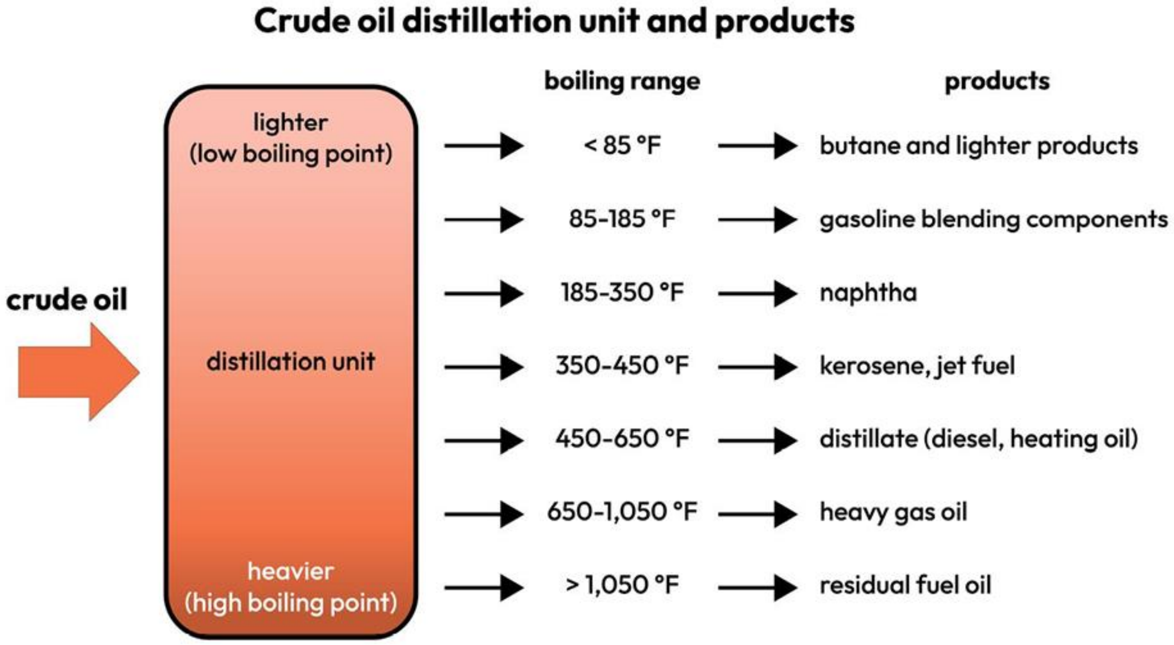
Crude oil distillation unit and products
crude oil
lighter (low boiling point)
distillation unit
heavier (high boiling point)
boiling range
products
< 85 °F
butane and lighter products
85-185 °F
gasoline blending components
185-350 °F
naphtha
350-450 °F
kerosene, jet fuel
450-650 °F
distillate (diesel, heating oil)
650-1,050 °F
heavy gas oil
> 1,050 °F
residual fuel oil

Figure 3. Crude oil basic boiling range with the help of wrangler to predict product type.

1.3 What Is AWS And Why Is It Used?

AWS stands for Amazon Web Services, It is an expanded cloud computing platform provided by Amazon Company. AWS provides a wide range of services with a pay-as-per-use pricing model over the Internet such as Storage, Computing power, Databases, Machine Learning services, and much more. AWS facilitates for both businesses and individual users with effectively hosting the applications, storing the data securely, and making use of a wide variety of tools and services improving management flexibility for IT resources. Then providing Simple Storage Service (Amazon S3) revolutionized with scalable management of Storage. Coming up with effective compute and storage services and providing them rental basis helped many startup companies and users with the cost of manual Hardware Infrastructure setup. Introducing the concept of server less computing with AWS lambda services enhanced its business globally. It came up with services like Elastic Beanstalk made the deployment of applications much easier bringing large audiences. AWS always came with diverse array of services offering with technical innovations, updated services with current trends. AWS has emerged as a powerhouse in the world of Cloud Computing.

2. How AWS Works?

AWS comes up with its own network infrastructure on establishing the datacenters in different regions mostly all over the world. Its global Infrastructure acts as a backbone for operations and services provided by AWS. It facilitates the users on creating secure environments using Amazon VPCs (Virtual Private Clouds). Essential services like Amazon EC2 and Amazon S3 for utilizing the compute and storage service with elastic scaling. It supports the dynamic scaling of the applications with the services such as Auto Scaling and Elastic Load Balancing (AWS ELB). It provides a good user-friendly AWS Management Console facilitating seamless configuration and management of AWS services to the Users. Its Architecture ensures high availability, fault tolerance making AWS as a versatile powerful Cloud Computing Platform.

2.1 AWS Fundamentals

In the Journey of AWS, understanding the key concepts such as Regions, Availability Zones, Global Network Infrastructure, etc.is crucial. The fundamentals of AWS keep on maintaining the applications reliable and scalable with services globally with coming to a strategic deployment of resources for optimal performance and resilience. The following are the some of the main fundamentals of AWS:

a) **Regions:**
AWS provide the services with respective division of regions. The regions are divided based on geographical areas/locations and will establish data centers. Based on need and traffic of users, the scale of data centers is depended to facilitate users with low-latencies of services.

b) **Availability Zones (AZ):**
To prevent the Data centers for the Natural Calamities or any other disasters. The Datacenters are established as sub sections with isolated locations to enhance fault tolerance and disaster recovery management.

c) **Global Network Infrastructure:**
AWS ensures the reliability and scalability of services through setting up its own AWS Network Infrastructure globally. It helps in better management of data transmissions for optimized performance and security reliance.

3. Top AWS Service

In the rapid revolution of Cloud Computing, AWS facilitates with wide variety of services respect to the fields and needs. The following are the top AWS services that are in wide usage:

I. **Amazon EC2 (Elastic Compute Cloud):** It provides the Scalable computing power via cloud allowing the users to run applications and manage the workloads over their remotely.

II. **Amazon S3 (Simple Storage Service):** It offers scalable object Storage as a Service with high durability for storing and retrieving any amount of data.

III. **AWS Lambda:** It is a service in Server less Architecture with Function as a Service facilitating server less computing i.e., running the code on response to the events, the background environment management of servers is handled by AWS automatically. It helps the developers to completely focus on the logic of code build.

IV. **Amazon RDS (Relational Database Service):** This is an AWS service that simplifies the management of database providing high available relational databases in the cloud.

V. **Amazon VPC (Virtual Private Cloud):** It enables the users to create isolated networks with option of public and private expose within the AWS cloud, providing safe and adaptable configurations of their resources.

3.1. Advantages of Amazon Web Services

- AWS allows you to easily scale your resources up or down as your needs change, helping you to save money and ensure that your application always has the resources it needs.
- AWS provides a highly reliable and secure infrastructure, with multiple data centers and a commitment to 99.99% availability for many of its services.
- AWS offers a wide range of services and tools that can be easily combined to build and deploy a variety of applications, making it highly flexible.
- AWS offers a pay-as-you-go pricing model, allowing you to only pay for the resources you actually use and avoid upfront costs and long-term commitments.

3.2. Disadvantages of Amazon Web Services

- AWS can be complex, with a wide range of services and features that may be difficult to understand and use, especially for new users.
- AWS can be expensive, especially if you have a high-traffic application or need to run multiple services. Additionally, the cost of services can increase over time, so you need to regularly monitor our spending.
- While AWS provides many security features and tools, securing your resources on AWS can still be challenging, and you may need to implement additional security measures to meet your specific requirements.
- AWS manages many aspects of the infrastructure, which can limit your control over certain parts of your application and environment.

4. Applications of AWS

The AWS services are using by both startup and MNC companies as per their use case. The startup companies are using overcome hardware infrastructure cost and applications deployments effectively with cost and performance. Whereas large scale companies are using AWS cloud services for the management of their Infrastructure to completely focus on the development of products widely. The following the Real-world industrial use-cases of AWS services:

- **Netflix:** The Large streaming gain using AWS for the storage and scanning of the applications for ensuring seamless content delivery with low latency without interruptions to millions of users globally.
- **Airbnb:** By utilizing AWS, Airbnb manages the various workloads and provides insurable and expandable infrastructure for its virtual marketplace and lodging offerings.
- **NASA’s Jet Propulsion Laboratory:** It takes the help of AWS services to handle and analyze large-scale volumes of data related to vital scientific research missions and space exploration.
- **Capital One:** A financial Company that is utilizing AWS for its security and compliance while delivering innovative banking services to its customers.

5. AWS Global Infrastructure

The AWS global infrastructure is massive and is divided into geographical regions. The geographical regions are then divided into separate availability zones. While selecting the geographical regions for AWS, three factors come into play.

- Optimizing Latency
- Reducing cost
- Government regulations (Some services are not available for some regions)

Each region is divided into at least two availability zones that are physically isolated from each other, which provides business continuity for the infrastructure as in a distributed system. If one zone fails to function, the infrastructure in other availability zones remains operational. The largest region North Virginia (US-East), has six availability zones. These availability zones are connected by high-speed fiber-optic networking.

There are over 100 edge locations distributed all over the globe that are used for the Cloud Front (content delivery network). Cloud Front can cache frequently used content such as images and videos (live streaming videos also) at edge locations and distribute it to edge locations across the globe for high-speed delivery and low latency for end-users. It also protects from DDOS attacks.

5.1. AWS Management Console

The AWS management console is a web-based interface to access AWS. It requires an AWS account and also has a smartphone application for the same purpose. So when you sign in for first time, you see the console home page where you see all the services provided by AWS. Cost monitoring is also done through the console.AWS resources can also be accessed through various Software Development Kits (SDKs), which allows the developers to create applications as AWS as its backend. There are SDKs for all the major languages(e.g., JavaScript, Python, Node.js, .Net, PHP, Ruby, Go, C++). There are mobile SDKs for Android, iOS, React Native, Unity. AWS can also be accessed by making HTTP calls using the AWS-API. AWS also provides a AWS Command Line Interface (CLI) for remotely accessing the AWS and can implement scripts to automate many processes. This Console is also available as an app for Android and iOS. For mobile apps, you can simply download AWS console app.

AWS Cloud Computing Models

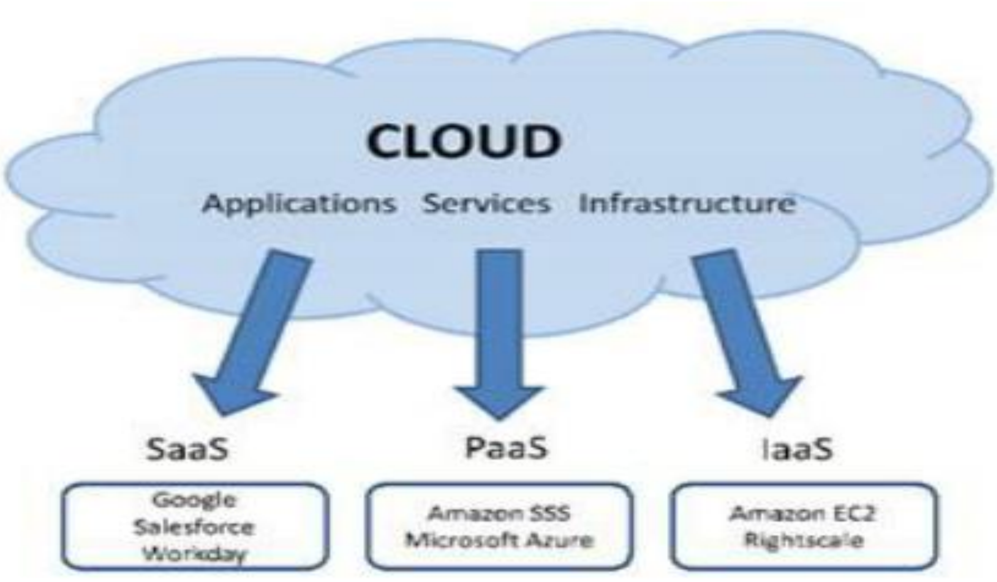

Figure 4. Infrastructure of AWS for Cloud computing

There are three cloud computing models available on AWS.

1. Infrastructure as a Service (IaaS):
It is the basic building block of cloud IT. It generally provides access to data storage space, networking features, and computer hardware (virtual or dedicated hardware). It is highly flexible and gives management controls over the IT resources to the developer. For example, VPC, EC2, EBS.

2. Platform as a Service (PaaS):
This is a type of service where AWS manages the underlying infrastructure (usually operating system and hardware). This helps the developer to be more efficient as they do not have to worry about undifferentiated heavy lifting required for running the applications such as capacity planning, software maintenance, resource procurement, patching, etc., and focus more on deployment and management of the applications. For example, RDS, EMR, Elastic Search.

3. Software as a Service (SaaS):
It is a complete product that usually runs on a browser. It primarily refers to end-user applications. It is run and managed by the service provider. The end-user only has to worry about the application of the software suitable to its needs. For example, Saleforce.com, Web-based email, Office 365.

5.2 Using AWS Cloud-Formation with layers

You can use AWS Cloud Formation to create a layer and associate the layer with your Lambda function. The following example template creates a layer named my-lambda-layer and attaches the layer to the Lambda function using the Layers property.

```
Description: Cloud Formation Template for Lambda Function with Lambda Layer
   Resources:
    MyLambdaLayer:
     Type: AWS: Lambda::LayerVersion
    Properties:
      LayerName: my-lambda-layer
      Description: My Lambda Layer
    Content:
       S3Bucket: my-bucket
       S3Key: my-layer.zip
    CompatibleRuntimes:
       - python3.9
       - python3.10
       - python3.11
    MyLambdaFunction:
     Type: AWS-Lambda:Function
     Properties:
      Function-Name: my-lambda-function
      Runtime: python3.9
      Handler: index_handler
      Timeout: 10
     Policies:
       - AWSLambdaBasicExecutionRole
       - AWSLambda_ReadOnlyAccess
       - AWSXrayWriteOnlyAccess
     Layers:
       - !Ref MyLambdaLayer
```

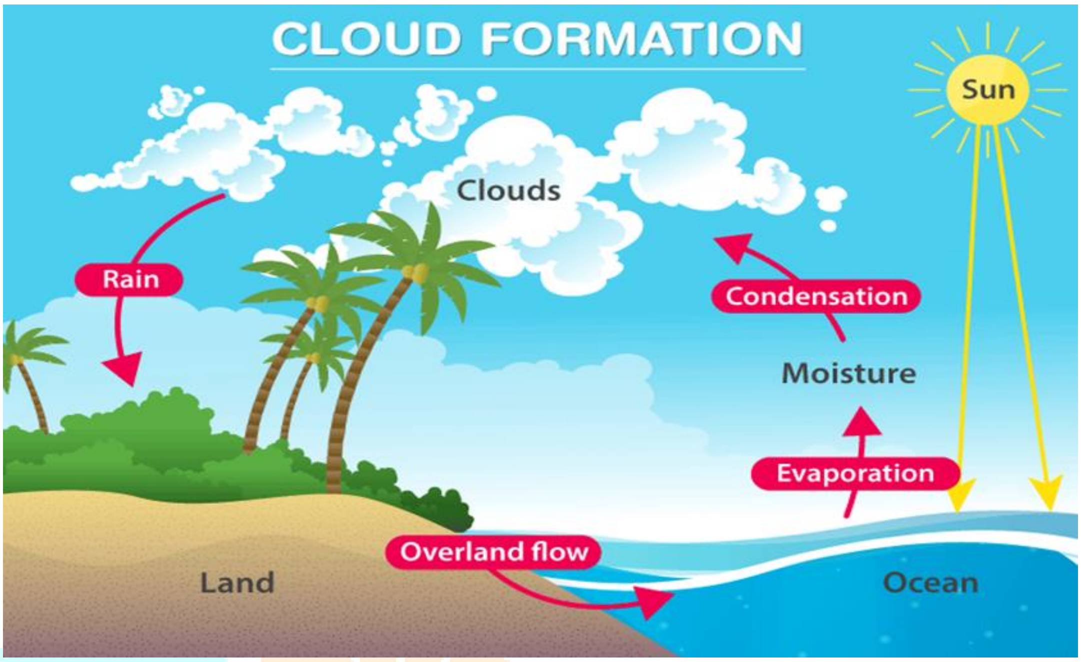

Figure 5. Cloud Information life cycle.

5.3 Building layers

You can use AWS SAM to build custom layers. For information about layers, see AWS Lambda layers in the AWS Lambda Developer Guide. To build a custom layer, declare it in your AWS Server less Application Model (AWS SAM) template file and include a Metadata resource attribute section with a Build Method entry. Valid values for Build Method are identifiers for an AWS Lambda runtime, or make file. Include a Build Architecture entry to specify the instruction set architectures that your layer supports. Valid values for Build Architecture are Lambda instruction set architectures.

If you specify make file, provide the custom make file, where you declare a build target of the form build-layer-logical-id that contains the build commands for your layer. Your make file is responsible for compiling the layer if necessary, and copying the build artifacts into the proper location required for subsequent steps in your workflow. The location of the make file is specified by the Content Uri property of the layer resource, and must be named Make file. When you include the Metadata resource attribute section, you can use the same build command to build the layer, both as an independent object, or as a dependency of an AWS Lambda function.

- **As an independent object.** You might want to build just the layer object, for example when you're locally testing a code change to the layer and don't need to build your entire application. To build the layer independently, specify the layer resource with the same build layer-logical-id command.
- **As a dependency of a Lambda function.** When you include a layer's logical ID in the Layers property of a Lambda function in the same AWS SAM template file, the layer is a dependency of that Lambda function. When that layer also includes a Metadata resource attribute section with a Build-Method entry, you build the layer either by building the entire application with the same build command or by specifying the function resource with the same build function-logical-id command.

Template example 1: Build a layer against the Python 3.9 runtime environment

The following example AWS SAM template builds a layer against the Python 3.9 runtime environment.

```
Resources:
MyLayer:
Type: AWS::Serverless::LayerVersion
Properties:
ContentUri: my_layer
CompatibleRuntimes:
- python3.9
Metadata:
BuildMethod: python3.9
# Required to have AWS SAM build this layer
```

Template example 2: Build a layer using a custom makefile

The following example AWS SAM template uses a custom makefile to build the layer.

```
Resources:
MyLayer:
Type: AWS::Serverless::LayerVersion
Properties:
ContentUri: my_layer
```

```
CompatibleRuntimes:
- python3.8
Metadata:
BuildMethod: makefile
```

The following makefile contains the build target and commands that will be executed. Note that the ContentUri property is set to my_layer, so the makefile must be located in the root of the my_layer subdirectory, and the filename must be Makefile. Note also that the build artifacts are copied into the python/ subdirectory so that AWS Lambda will be able to find the layer code.

```
build-MyLayer:
  mkdir -p "$(ARTIFACTS_DIR)/python"
  cp *.py "$(ARTIFACTS_DIR)/python"
 python -m pip install -r requirements.txt -t "$(ARTIFACTS_DIR)/python"
```

Example same build commands

The following sam build commands build layers that include the Metadata resource attribute sections.

```
# Build the 'layer-logical-id' resource independently
$ sam build layer-logical-id
# Build the 'function-logical-id' resource and layers that this function depends on
$ sam build function-logical-id
# Build the entire application, including the layers that any function depends on
$ sam build
```

6. CONCLUSION

Data wrangling is the process of cleaning, transforming, and organizing raw, messy, or unstructured data into a structured format. It involves processes such as data cleaning, data integration, data transformation, and data enrichment to ensure that the data is accurate, consistent, and suitable for analysis. Data Wrangling on AWS equips you with the knowledge to reap the full potential of AWS data wrangling tools. First, you'll be introduced to data wrangling on AWS and will be familiarized with data wrangling services available in AWS. You'll understand how to work with AWS Glue Data Brew, AWS data wrangler, and AWS Sage maker. Next, you'll discover other AWS services like Amazon S3, Redshift, Athena, and Quick sight. Additionally, you'll explore advanced topics such as performing Pandas data operation with AWS data wrangler, optimizing ML data with AWS Sage Maker, building the data warehouse with Glue Data Brew, along with security and monitoring aspects. By the end of this book, you'll be well-equipped to perform data wrangling using AWS services.

REFERENCES

1. Cline Don, Yueh Simon and Chapman Bruce, Stankov Boba, Al Gasiewski, and Masters Dallas, Elder Kelly, Richard Kelly, Painter Thomas H., Miller Steve, Katzberg Steve, Mahrt Larry, (2009), NASA Cold Land Processes Experiment (CLPX 2002/03): Airborne Remote Sensing.
2. A. Fatima, N. Nazir, and M. G. Khan, "Data Cleaning in Data Warehouse: A Survey of Data Pre-processing Tech- niques and Tools," Int. J. Inf. Technol. Comput. Sci., vol. 9, no. 3, pp. 50–61, 2017.
3. Kandel Sean, Paepcke Andreas, Hellersteiny Joseph and Heer Jeffrey (2011), Wrangler: Interactive Visual Specifi- cation of Data Transformation Scripts, ACM Human Fac- tors in Computing Systems (CHI) ACM 978-1-4503- 0267-8/11/05.
4. Chaudhuri. S and Dayal. U (1997), an overview of data warehousing and OLAP technology. In SIGMOD Record
5. (2001) "Potter's Wheel: An Interactive Data Cleaning Sys- tem", Proceedings of the 27th VLDB Conference.
6. Ahuja.S, Roth.M, Gangadharaiah R, Schwarz.P and Bas- tidas.R, (2016), "Using Machine Learning to Accelerate Data Wrangling", IEEE 16th International Conference on Data Mining Workshops (ICDMW), 2016, Barcelona, Spain, pp. 343-349.doi:10.1109/ICDMW.2016.0055.
7. Data wrangling platform (2017) publication, www.trifacta.com. [Online] Available: https://www.trifacta.com/products/architecture//, [Ac- cessed on: 01 May 2017].
8. Norman D.A, (2013), Text book on "The Design of Eve- ryday Things, Basic Books", [Accessed on 12 April 2017].
9. Anderson, James. "Cloud Computing: A Comprehensive Security Framework." *Journal of Computer Security*, vol. 28, no. 1, 2017, pp. 89-104.
10. Chang, Li and Gupta, Ananya. "Data Breaches in Cloud Computing: Trends and Mitigation Strategies." *International Journal of Information Management*, vol. 35, no. 6, 2018, pp. 672-679.
11. Brown, Karen and Williams, Robert. "Securing Cloud Data: Current Trends and Future Directions." *Journal of Cloud Security*, vol. 12, no. 2, 2019, pp. 145-162.
12. Chen, Mei and Lee, Wei. "A Comparative Analysis of Cloud Security Measures." *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 3, 2020, pp. 458-471.
13. Dhillon, Gurpreet and Moores, Trevor. "Data Encryption in Cloud Computing: A Comprehensive Review." *Journal of Information Privacy & Security*, vol. 35, no. 4, 2021, pp. 378-394.
14. Smith, John. "Cloud Security: Challenges and Opportunities." *Journal of Information Security*, vol. 30, no. 2, 2018, pp. 45-62.
15. Wang, Li and Zhang, Wei. "Securing Information Stockpiling in Cloud Computing: A Comprehensive Review." *International Journal of Cybersecurity and Privacy*, vol. 15, no. 3, 2019, pp. 112-128
16. Johnson, Emily and Davis, Michael. "Cyber Threats in Cloud Computing: A Comprehensive Analysis." *Journal of Computer Security*, vol. 25, no. 4, 2020, pp. 321- 335.
17. Gupta, Rajesh and Patel, Meera. "Enhancing Cloud Security: Emerging Trends and Technologies." *International Conference on Cybersecurity and Data Protection*, 2017, pp. 87-94.

18. Li, Xia and Chen, Wei. "Blockchain Technology for Cloud Security: Opportunities and Challenges." *IEEE Transactions on Cloud Computing*, vol. 8, no. 3, 2019, pp. 752-761.
19. Ristenpart, T., Tromer, E., Shacham, H., & Savage, S. (2009). Hey, You, Get Off of My Cloud: Exploring Information Leakage in Third-Party Compute Clouds. ACM Conference on Computer and Communications Security.
20. Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing. National Institute of Standards and Technology (NIST) Special Publication 800-145.
21. Whitman, M. E., & Mattord, H. J. (2018). Management of Information Security. Cengage Learning.
22. Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.
23. SANS Institute. (2021). Incident Handling and Response. SANS Institute InfoSec Reading Room.