# ADF CHEATSHEET

BY - SHUBHAM WADEKAR

- **Pipeline**: A workflow unit in ADF that groups multiple activities to carry out a specific data integration task.
- **Activity**: Represents a single operation within a pipeline, such as copying data between sources.
- **Dataset**: Defines metadata for data stored externally; activities rely on datasets to interact with the data.
- **Linked Service**: Serves as a connection definition to external compute or storage systems.
- **Integration Runtime (IR)**: The compute infrastructure used by ADF for executing activities — ADF itself doesn't provide native storage.
- **Debug Mode**: Allows testing pipelines interactively in the ADF UI without publishing — it treats all resources as it would in production.
- **Copy Data Wizard**: A step-by-step UI tool for quickly setting up pipelines that perform data copy operations, though it's rarely used in enterprise setups.
- **Azure Storage**: Microsoft's managed cloud storage platform, enabling scalable data storage.
- **Storage Account**: A required container to access and use Azure Storage services.
- **Storage Access Key**: Used to authenticate access to a storage account; can be managed via the Azure portal.
- **Blob Storage**: One of Azure's storage offerings, designed to store large amounts of unstructured data.
- **Container**: Logical subdivisions within blob storage accounts where blobs (files) are stored — containers are not nested.
- **Azure Storage Explorer**: A GUI application that helps manage Azure Storage resources, available both as a desktop and web app.
- **Bandwidth**: Refers to the volume of data entering or exiting Azure's infrastructure. Outbound traffic may incur costs (egress charges).
- **Unstructured File**: A file treated without a known schema (e.g., a binary blob). Copy activity processes these as raw binaries.
- **Structured File**: Tabular files such as CSV or Parquet with defined columns and rows.
- **Parquet**: A compact, column-based file format well-suited for storing and querying large datasets efficiently.
- **Semi-Structured File**: Files like JSON or XML that contain flexible, sometimes nested data structures.
- **Collection Reference**: Used during schema mapping in Copy activities to identify a specific nested collection being processed.
- **Sink**: The target or destination where transformed or copied data is written.
- **Interim Data Types**: ADF uses an intermediate format to map source and sink types during copy operations for broader compatibility.
- **Data Integration Unit (DIU)**: ADF's measurement of compute power, combining CPU, memory, and network resources. DIUs influence performance and cost.
- **Degree of Parallelism (DoP)**: The number of parallel threads a Copy activity uses. Manual configuration is possible but not usually recommended.

- **Azure SQL Database**: A fully managed cloud SQL solution offered by Azure.
- **Logical Server**: A virtual container grouping several Azure SQL DB instances for easier management.
- **Online Query Editor**: A browser-based tool to write and run queries against Azure-hosted databases.
- **Expression**: Runtime-evaluated statements used in pipelines to calculate or assign values dynamically.
- **Array**: A list of values accessible via an index; used for iteration and dynamic operations.
- **Dictionary**: A key-value pair collection used for referencing named elements.
- **Expression Builder**: A tool within ADF UX for authoring and validating expressions.
- **System Variable**: Predefined variables providing runtime metadata (e.g., pipeline run ID, trigger name).
- **User Variable**: Custom variables created within a pipeline for storing values like strings, booleans, or arrays.
- **Expression Functions**: ADF includes a rich set of functions (e.g., math, string, date, type conversions) for use in expressions.
- **Interpolated Strings**: String literals embedded with expressions evaluated during execution.
- **Placeholder Expression**: An embedded expression inside a string that resolves to a final value at runtime.
- **Escape Sequences**: To prevent the "@" character from being treated as an expression, use "@@" instead.
- **Stored Procedure Activity**: Executes SQL stored procedures with support for parameterized inputs.
- **Lookup Activity**: Queries external data sources and returns the results for use in downstream pipeline logic.
- **Set Variable Activity**: Assigns a new value to an existing user variable.
- **Append Variable Activity**: Adds a new item to an array-type variable.
- **Activity Dependency**: Defines execution order between pipeline activities, based on conditions like success or failure.
- **Activity Output**: The result from a pipeline activity, provided as a JSON object usable by downstream steps.
- **Breakpoint**: Enables stopping execution during debugging after a selected activity; not supported in live runs.
- **$$FILEPATH**: Inserts the file path of the incoming file into the dataset during a Copy activity — not usable in expressions.
- **$$COLUMN**: Duplicates a specific column in Copy activity — only for populating additional columns, not for logic expressions.
- **Additional Columns**: Let you add hardcoded or dynamic columns (via expressions or system variables) during copy.

- **Lineage Tracking**: Helps trace data's journey from source to destination to improve traceability.
- **Runtime Parameters**: Dynamic values substituted during execution — applicable at pipeline, dataset, or linked service level.
- **Optional Parameters**: Runtime parameters that have default values, making them optional at runtime.
- **Reusability**: Using parameters enhances component reusability by customizing behavior without duplicating resources.
- **Global Parameter**: Factory-wide constants referenced in expressions, using pipeline().globalParameters.ParamName.
- **Pipeline Parameter**: Runtime parameters scoped to individual pipelines, referenced via pipeline().parameters.ParamName.
- **Dataset Parameter**: Parameters available within dataset expressions, accessed as dataset().ParamName.
- **Linked Service Parameter**: Used to parameterize connections; syntax is linkedService().ParamName. Not always configurable via ADF UX — may require JSON editing.
- **Execute Pipeline Activity**: Triggers another pipeline within the same ADF instance.
- **Azure Key Vault**: A secure cloud store for managing credentials and secrets.
- **Secret**: Sensitive data stored in Key Vault, like access keys or passwords, referenced securely via names.
- **Service Principal**: An identity created for a service or application to allow secure access to Azure resources.
- **Managed Identity**: Azure-managed identity linked to ADF (or other services) used for secure authentication.
- **Access Policy**: Defines who/what can access a Key Vault and under what conditions.
- **Dependency Condition**: Defines whether downstream activities run based on the status of previous ones (success, failure, skipped).
- **Multiple Dependencies**: An activity waiting on several others will only proceed if all their conditions are met.
- **Leaf Activity**: A terminal activity in a pipeline that doesn't lead to any other activity.
- **Conditional Activities**: Includes If Condition and Switch — control flow elements based on runtime logic.
- **If Condition**: Runs one of two activity sets based on whether a condition is true or false.
- **Switch Activity**: Routes execution to one of several branches depending on the result of a string-evaluated expression.
- **Iteration Activities**: Includes ForEach and Until — used for repeating actions.
- **ForEach**: Loops through array elements and runs activities per item. Default execution is parallel.

- **Parallelism in ForEach**: Supports concurrent execution; care is needed to avoid state conflicts. Debug runs are always sequential.
- **Until Activity**: Repeats activities until a condition evaluates to true. Always runs at least once and never in parallel.
- **Nesting Restrictions**: You can't nest loops within loops or conditions within conditions. Use sub-pipelines as a workaround.
- **Iteration Breakpoints**: Not supported within loops or conditions in ADF UX.
- **Get Metadata Activity**: Extracts metadata from datasets (e.g., file size, existence). Misconfiguration can cause failures.
- **Fault Tolerance in Copy Activity**: Allows logging of failed rows without halting the entire data load.
- **Simulating Errors**: While ADF lacks a native "raise error" function, errors can be forced via bad casts or SQL statements like RAISERROR.
- **Apache Spark**: An open-source engine for distributed data processing, optimized for parallel computation across a cluster.
- **Databricks**: A cloud-based platform built on Spark, providing collaborative environments and enterprise-grade features.
- **Data Flows in ADF**: A visual design feature in ADF that enables data transformation using an underlying Spark engine (Databricks).
- **Data Flow Debug Mode**: When enabled, a temporary Databricks cluster is created to test data flows during development.
- **Time To Live (TTL)**: The idle time before a debug cluster shuts down automatically, defaulting to one hour.
- **Data Flow Activity**: Executes a data flow inside an ADF pipeline.
- **Data Flow Parameters**: Variables you define during testing in debug settings and pass values to during pipeline execution.
- **Data Flow Canvas**: The drag-and-drop design surface used to build and arrange transformations visually.
- **Transformation**: A step in a data flow that manipulates data — each transformation modifies the stream.
- **Output Stream Name**: A unique name assigned to each transformation within a data flow, used for referencing.
- **Inspect Tab**: Displays the schema details (input and output) for a specific transformation.
- **Data Preview Tab**: Shows sample output for a transformation during debug; also helps avoid cluster timeout.
- **Optimize Tab**: Adjusts data partitioning strategies used by Spark when executing transformations.
- **Source Transformation**: Begins a data flow by pulling data from a configured external source.
- **Sink Transformation**: Final step in a data flow that writes the result to an external system.

- **Data Flow Expression Language**: A custom expression language for transformations — different from pipeline expression syntax.
- **Data Flow Script**: The underlying code structure representing the transformation logic in JSON format.
- **Column Patterns**: Allow for applying transformations to multiple columns based on pattern-matching metadata.
- **Filter Transformation**: Filters incoming rows based on a condition; only matching rows pass through.
- **Lookup Transformation**: Works like a join; combines two data streams based on key relationships.
- **Derived Column Transformation**: Adds new fields to the stream by evaluating expressions.
- **Locals**: Variables within Derived Column transformations to simplify or reuse expressions.
- **Select Transformation**: Used to rename or drop columns from the data stream.
- **Aggregate Transformation**: Performs group-based operations such as sums or counts.
- **Exists Transformation**: Keeps or discards rows based on the presence of matching rows in another stream.
- **Templates**: Ready-made reusable pipeline or data flow designs for common patterns.
- **Template Gallery**: Built-in template library accessible from the ADF overview page.
- **External Activity**: Any activity that runs on compute outside ADF, like Databricks notebooks or SQL procedures.
- **Internal Activity**: Executes using ADF's own managed integration runtime.
- **Integration Runtime (IR)**: The engine behind activity execution; can be managed by Azure or self-hosted.
- **Dispatching**: ADF's process of allocating activities (especially external) to appropriate compute environments.
- **Azure Integration Runtime (Azure IR)**: A managed, serverless IR used for data flows and Copy activities. Handles transformation and movement.
- **AutoResolveIntegrationRuntime**: A default IR in every ADF instance that auto-selects compute location and cluster configuration.
- **Self-hosted IR**: An IR that runs on your own infrastructure. Used for connecting to on-premise systems or unsupported connectors.
- **Linked Self-hosted IR**: A shared IR configuration allowing other factories to reference an existing self-hosted IR.
- **Azure-SSIS IR**: Managed VMs provided by Azure to run SSIS packages as part of ADF workflows.
- **Web Activity**: Enables REST API calls within a pipeline — useful for integrations and triggering services.
- **Power Query in ADF**: An interactive, visual tool for shaping and preparing data. Based on Power Query used in Power BI and Excel.
- **Data Wrangling**: The process of exploring and transforming data interactively using Power Query.

- **Mashup**: A Power Query transformation script created in the visual editor.
- **M Language**: The functional language behind Power Query transformations. It's translated into data flow script at runtime.
- **Power Query Activity**: Executes a mashup created via Power Query in the ADF interface.
- **ARM Template**: JSON-based deployment file that describes the configuration of Azure resources — including ADF components.
- **Publish**: The action that deploys a pipeline from draft (UX) to production. Required to trigger pipeline runs via schedules or events.
- **Publish Branch**: A special branch (usually adf_publish) in Git that holds the published JSON and ARM templates.
- **Custom Azure Role**: A user-defined role with a tailored set of permissions, used when default Azure roles are insufficient.
- **Deployment Parameters**: Variables in ARM templates that allow customizing deployments for different environments.
- **Parameterization Template**: A JSON template used to flag which properties should be parameterized during deployment.
- **CI/CD (Continuous Integration/Continuous Delivery)**: Practice of automating code integration, testing, and deployment in Azure.
- **Azure Pipelines**: Azure DevOps service that automates builds, tests, and deployments through pipeline definitions.
- **Data Serialization Language**: A format for storing/transmitting structured data. Examples: XML, JSON, YAML.
- **YAML**: A clean, indentation-sensitive language often used to define DevOps pipelines in Azure.
- **Pipeline Task**: A unit of work in a DevOps pipeline, such as executing a script or deploying resources.
- **Pipeline Variable**: A variable declared in a DevOps pipeline — secret variables can store sensitive data securely.
- **Service Connection**: An authentication bridge to allow DevOps pipelines to interact with Azure using AAD credentials.
- **Feature Branch Workflow**: A Git branching strategy where each feature is developed independently before being merged.
- **Pull Request (PR)**: A GitHub/Azure DevOps request to merge feature code into a shared branch after review.
- **Az.DataFactory**: PowerShell module that provides cmdlets for working with ADF resources and operations.
- **Trigger**: A mechanism in ADF that initiates the execution of one or more pipelines based on defined conditions or schedules.
- **Trigger Run**: Represents a single instance of a trigger firing — may launch one or multiple pipeline executions depending on its configuration.
- **Trigger Start Time**: The moment from which a trigger begins to monitor or execute.

- **Trigger End Time**: The cutoff time after which the trigger stops running automatically.
- **Recurrence Pattern**: Defines a repeating time schedule (e.g., every 2 hours, daily at 9 AM) that controls when a trigger fires.
- **Schedule Trigger**: A time-based trigger driven by the system clock or specified recurrence interval.
- **Event-Based Trigger**: Fires in response to external events, such as the creation or deletion of a file in Azure Blob Storage.
- **Resource Provider**: Azure uses these to manage specific resource types; ADF depends on registered resource providers to work with external services.
- **Azure Event Grid**: The event distribution backbone of Azure — ADF uses it to consume blob storage events for event-based triggering.
- **Tumbling Window Trigger**: A time-windowed trigger type that divides time into contiguous slices, each corresponding to a unique pipeline execution window. Supports retries, dependencies, and concurrency control.
- **Pipeline Run Overlap**: Occurs when a trigger launches a new run before the previous one completes. Tumbling windows with self-dependencies can prevent overlaps.
- **Reusable Trigger**: A trigger (schedule or event-based) that can be linked to multiple pipelines. Tumbling windows are restricted to a single pipeline.
- **Trigger-Scoped System Variables**: Special system variables accessible in triggers, some of which vary depending on trigger type.
- **Azure Logic Apps**: The internal automation engine behind ADF triggers; used for orchestration and workflow execution.
- **Trigger Publishing**: Triggers must be published to become active; they do not function in debug mode.
- **Pipeline Annotation**: A custom tag added to a pipeline; shows up in execution logs and helps with grouping or filtering log data.
- **Trigger Annotation**: Similar to pipeline annotations but applied to triggers for logging and organization purposes.
- **Activity User Property**: Custom key-value metadata added to an activity, visible in logs for tracking and reporting. Copy activity includes auto-generated properties for source and destination identifiers.
- **Azure Monitor**: Azure's centralized service for tracking, analyzing, and responding to metrics and logs across services.
- **Metric**: Time-series numerical values automatically captured for system components, often visualized or queried.
- **Log Analytics**: A part of Azure Monitor focused on aggregating and querying logs for diagnostics and insights.
- **Log Analytics Workspace**: A designated environment that collects and stores logs and metrics for long-term query and analysis.
- **Diagnostic Setting**: Configures which logs and metrics from an Azure resource should be sent to a monitoring destination (e.g., a Log Analytics workspace).

- **Kusto**: A powerful query language used with Azure Monitor and Azure Data Explorer to analyze log and telemetry data.
- **Tabular Expression Statement**: The primary type of Kusto query — returns data in table form; every query must end with one.
- **Log Analytics Workbook**: A document-like interface that combines Kusto queries, visualizations, and commentary to create monitoring dashboards or reports.
- **Azure Data Explorer**: A highly scalable analytics platform that supports real-time querying of massive datasets using Kusto.
- **Alerts**: Notifications triggered when certain log patterns or metric thresholds are met, helping to monitor systems proactively.
- **Alert Rule**: Defines what condition should raise an alert, where to look for the signal, and what action should be taken.
- **Signal**: The data point (metric, log, or custom query result) used to determine whether an alert condition has been satisfied.