**AWS Basics:**

|  |  |  |
| --- | --- | --- |
| AWS – EC2 | AWS – extract | AWS -S3 |
| AWS -Bedrock | AWS – Sage maker | AWS – Code commit |
| AWS – Code build | AWS – Code deploy | AWS – Cloud watch |
| Jenkin | Docker |  |

1. **AWS – EC2.**

Amazon EC2 (Elastic Compute Cloud) is a web service that provides resizable compute capacity in the cloud, allowing you to run virtual machines (called instances) on demand, offering flexibility and scalability for various applications

**What it is:**

EC2 is a core component of Amazon Web Services (AWS) that enables you to rent virtual computers (instances) to run your applications on the AWS cloud infrastructure.

**Key Features:**

* Scalability: You can easily scale your computing resources up or down based on demand, paying only for what you use.
* Flexibility: Choose from a wide variety of instance types (different CPU, memory, storage, and networking configurations) to optimize for your specific workload.
* Control: You have control over the operating system, storage, and networking, allowing you to build and run virtually any application.
* Security: EC2 offers robust security features, including security groups and network firewalls, to protect your instances and data.

**Use Cases:**

* Web servers and web applications.
* Database servers.
* Batch processing and data analysis.
* High-performance computing (HPC).
* Application development and testing.

**Pricing:**

EC2 offers various pricing models, including on-demand, reserved, and spot instances, allowing you to choose the best option for your needs and budget.

1. **AWS – extract.**

AWC extract" can refer to a few things, depending on the context. It could mean extracting data from a Siemens Active Workspace Client (AWC) system, or it could refer to "Acceptance, Waiver, and Consent" (AWC) documents.

Here's a breakdown of the possible meanings:

***Siemens Active Workspace Client (AWC) Data Extraction:***

* Context:

In the context of Siemens' Teamcenter software and its Active Workspace Client (AWC), "AWC extract" likely refers to the process of exporting data, stylesheets, preferences, or other items from the AWC system.

* Tools and Utilities:

You can use utilities like admin\_data\_export to export stylesheets, preferences, and other items.

* Locations:

You can find code in awc2\\stage folder, custom modules, and source code repositories.

* Commands:

You can also use commands like export\_uiconfig, export\_wsconfig, import\_uiconfig, and import\_wsconfig to export and import data.

1. **AWS – S3.**

Amazon Simple Storage Service (S3) is a cloud object storage service offered by Amazon Web Services (AWS) providing scalable, reliable, fast, and inexpensive data storage for a variety of use cases.

Here's a more detailed explanation:

***Key Features and Concepts:***

**Object Storage:**

S3 stores data as "objects" (files with associated metadata) within "buckets" (containers).

**Scalability and Durability:**

S3 is designed for high scalability, allowing users to store and retrieve any amount of data, and offers industry-leading data durability (11 nines).

**Data Availability:**

S3 provides high availability, ensuring that your data is accessible when you need it.

**Security:**

S3 is secure, private, and encrypted by default, with various security features and auditing capabilities.

**Storage Classes:**

S3 offers different storage classes optimized for various use cases, including frequent access, infrequent access, and archival storage.

**Use Cases:**

* S3 can be used for diverse applications, including:
* Web Hosting: Storing static website content.
* Data Storage: Storing and retrieving large datasets.
* Backups and Disaster Recovery: Creating backups and storing data for disaster recovery purposes.
* Data Lakes: Building data lakes for analytics workloads.
* Content Distribution: Distributing content to users efficiently.

**Integration with other AWS Services:**

S3 integrates seamlessly with other AWS services, making it a powerful tool for building and managing cloud applications.

**S3 Tables:**

S3 Tables are specifically optimized for analytics workloads, resulting in up to 3x faster query performance compared to unmanaged Iceberg tables and up to 10x higher transactions per second compared to Iceberg tables stored in general purpose S3 buckets.

1. **AWS – Bedrock.**

Amazon Bedrock is a fully managed service offered by Amazon Web Services (AWS) that simplifies building and scaling generative AI applications by providing access to a variety of foundation models (FMs) and tools for developers to create AI applications securely and privately.

Here's a more detailed breakdown:

**Key Features and Benefits:**

**Access to Foundation Models:**

Bedrock provides access to a selection of high-performing FMs from leading AI companies like Anthropic, Cohere, Meta, Mistral AI, Stability AI, and AI21 Labs, all accessible through a single API.

**Simplified Generative AI Development:**

Bedrock simplifies the development process by offering a serverless environment, eliminating the need to manage infrastructure and allowing developers to focus on building applications.

**Secure and Responsible AI:**

Bedrock is designed with security and privacy in mind, offering tools and features to ensure the responsible use of AI, including safeguards to prevent harmful content and factual errors.

**Model Customization:**

Developers can customize foundation models privately with their own data using techniques like fine-tuning and retrieval-augmented generation (RAG).

**Integration with AWS Services:**

Bedrock integrates well with other AWS services, making it easy to deploy generative AI capabilities into existing applications.

**Bedrock Marketplace:**

Amazon Bedrock Marketplace allows developers to discover, test, and use over 100 popular, emerging, and specialized foundation models alongside the current selection of industry-leading models in Amazon Bedrock.

**Bedrock Agents:**

Bedrock Agents are a feature that allows developers to create managed agents that execute complex business tasks, such as booking travel, processing insurance claims, creating ad campaigns, and managing inventory, all without writing any code.

**Bedrock Guardrails:**

Amazon Bedrock Guardrails provides configurable safeguards to help safely build generative AI applications at scale.

1. **AWS – Sage maker.**

Amazon SageMaker is a fully managed machine learning (ML) service from AWS that simplifies the process of building, training, and deploying ML models, automating many of the tasks involved in the ML lifecycle.

Here's a breakdown of what SageMaker is used for:

Building and Training ML Models:

SageMaker provides tools and algorithms to help data scientists and developers build, train, and fine-tune ML models, whether using pre-built algorithms or their own frameworks and code.

**Deploying Models:**

SageMaker facilitates the deployment of trained models into production-ready, hosted environments, making it easier to put ML models into real-world applications.

**Automating ML Lifecycle:**

SageMaker automates many of the tasks involved in the ML lifecycle, including data preparation, model training, hyperparameter tuning, and model deployment, streamlining the process and accelerating the overall ML workflow.

**Supporting Various ML Frameworks:**

SageMaker supports popular ML frameworks like TensorFlow, PyTorch, and MXNet, allowing users to leverage their preferred tools and libraries.

**Integrated Development Environment (IDE):**

SageMaker Studio provides a unified, web-based IDE for all stages of the ML workflow, from data preparation and model development to training, tuning, and deployment.

**Data Processing and Analytics:**

SageMaker integrates with other AWS services like Amazon S3, AWS Glue, and Amazon EMR for data processing and analytics, enabling users to access and prepare data from various sources.

**Collaboration and Version Control:**

SageMaker Studio supports collaboration and version control through Git integration, allowing teams to work together on ML projects and track changes.

**Scalability and Cost-Effectiveness:**

SageMaker is designed to be scalable ax`nd cost-effective, allowing users to train and deploy models on a large scale without worrying about infrastructure management.

1. **AWS – Code commit.**

AWS CodeCommit is a fully managed, secure, and highly scalable source control service that allows you to privately store and manage Git repositories in the AWS Cloud, eliminating the need to operate your own source control system.

Here's a more detailed explanation:

**What it is:**

CodeCommit is a service provided by Amazon Web Services (AWS) that enables developers to store and collaborate on their code repositories.

**Key Features:**

* **Secure and Private**: It allows you to host private Git repositories, ensuring your code is protected.
* **Managed Service**: AWS handles the infrastructure and maintenance, so you don't have to worry about scaling or managing the source control system.
* **Scalable:** It can handle large repositories and high traffic loads.
* **Git Integration:** It works seamlessly with your existing Git tools and workflows.
* **Compliance:** CodeCommit is in scope with many compliance programs, including HIPAA and PCI DSS.

**Use Cases:**

* Storing and managing source code, documents, and binary files.
* Collaborating on code with other developers.
* Integrating with other AWS services, such as CodeBuild and CodeDeploy.

**Benefits:**

* Reduced Operational Burden: You don't need to manage the infrastructure or maintenance of a source control system.
* Improved Security: Private repositories and AWS security features ensure your code is protected.
* Enhanced Collaboration: Easy access to and collaboration on code repositories.
* Simplified CI/CD: Seamless integration with other AWS services for continuous integration and continuous delivery.

1. **AWS – Code build.**

AWS CodeBuild is a fully managed, cloud-based continuous integration service that compiles source code, runs tests, and produces software packages ready for deployment, eliminating the need to manage your own build servers.

Here's a more detailed explanation:

**Fully Managed Service:**

AWS handles the infrastructure, provisioning, scaling, and maintenance of build servers, allowing you to focus on your code.

**Continuous Integration:**

CodeBuild integrates seamlessly with other AWS services like AWS CodeCommit and AWS CodePipeline, enabling automated builds and tests on every code change.

**Build Process:**

You specify the location of your source code and build settings, and CodeBuild handles the rest, including compiling, testing, and packaging your code.

**Prepackaged Build Environments:**

CodeBuild provides prepackaged build environments for popular programming languages and build tools, or you can customize them with your own tools.

**Scalability:**

CodeBuild automatically scales to meet your build needs, ensuring that builds don't wait in a queue.

**Cost-Effective:**

You only pay for the resources you use, with no upfront costs or minimum fees.

**Use Cases:**

* Automate continuous integration and delivery (CI/CD) pipelines.
* Remove the complexity of managing build servers.
* Build source code hosted on GitHub.
* Run existing Jenkins build jobs on CodeBuild.

**Build Specification File:**

CodeBuild uses build specification files (often in YAML format) to define build commands, environment variables, and other settings.

1. **AWS – Code deploy.**

AWS CodeDeploy is a fully managed deployment service that automates application deployments to various compute services like Amazon EC2, on-premises servers, and AWS Lambda, simplifying and streamlining the deployment process.

Here's a more detailed explanation:

**Automated Deployments:**

CodeDeploy automates the process of deploying applications, reducing manual intervention and potential errors.

**Compute Services:**

It supports deployments to Amazon EC2 instances, on-premises servers, and AWS Lambda functions, providing flexibility for different deployment scenarios.

**Rapid Releases:**

CodeDeploy enables developers to rapidly release new features and updates to their applications, improving agility and time-to-market.

**Downtime Avoidance:**

It helps avoid downtime during deployments through techniques like rolling updates and deployment health tracking.

**Centralized Control:**

CodeDeploy provides centralized control over deployments through the AWS Management Console, AWS CLI, SDKs, or APIs, allowing you to launch, control, and monitor deployments.

**Platform and Language Agnostic:**

It's platform and language agnostic, meaning it can be used with any application and setup code, allowing you to reuse existing setup code.

**Deployment Groups:**

You can specify one or more deployment groups for a CodeDeploy application. Each application deployment uses one of its deployment groups. The deployment group contains settings and configurations used during the deployment.

**Deployment Configurations:**

A deployment configuration is a constraint that determines how a deployment progresses through a deployment group.

**Deployment Types:**

CodeDeploy supports multiple deployment types, including in-place, canary, and blue/green deployments.

**Monitoring and Rollbacks:**

You can monitor the health of your deployments and automatically roll back updates if necessary.

**Integration with other AWS services:**

CodeDeploy is designed to be used as a building block service and is intended to be used in conjunction with other AWS deployment services such as AWS CodeStar, AWS CodePipeline, and other AWS Developer Tools.

1. **AWS – Cloud watch.**

Amazon CloudWatch is a monitoring and management service offered by AWS that allows you to collect and track metrics, collect and monitor log files, set alarms, and respond to changes in your AWS resources, providing insights into the operational health of your applications and infrastructure.

Here's a more detailed explanation:

**Key Features and Functionality:**

**Monitoring AWS Resources:**

CloudWatch enables real-time monitoring of various AWS resources, including EC2 instances, EBS volumes, Elastic Load Balancing, and RDS instances, automatically collecting metrics like CPU utilization, latency, and request counts.

**Collecting and Storing Logs:**

CloudWatch Logs allows you to collect and store logs from your resources, applications, and services in near real-time, enabling you to centralize logs from all your systems and applications.

**Setting Alarms:**

You can set alarms based on CloudWatch metrics to be notified when a system or service reaches a predefined threshold, such as errors or high usage.

**Dashboards:**

CloudWatch dashboards provide a customizable view of your metrics and alarms, allowing you to monitor your resources in a single view, even those spread across different regions.

**Insights and Investigations:**

CloudWatch helps you investigate operational issues in your environment, offering insights into the performance and health of your infrastructure and applications.

**Integration with other AWS Services:**

CloudWatch integrates with other AWS services, such as AWS Lambda, allowing you to monitor Lambda functions and capture logs for all requests handled by your functions.

**Cost-Effective Monitoring:**

CloudWatch provides cost-effective real-time visibility into your metrics with no deployment and no maintenance.

**Cross-Account Observability:**

CloudWatch allows you to monitor resources across multiple AWS accounts.

**Container Insights:**

CloudWatch provides enhanced observability for containers.

1. **Jenkin.**

Jenkins is an open-source automation server, primarily used for continuous integration and continuous delivery (CI/CD), that helps automate tasks related to building, testing, and deploying software.

Here's a more detailed explanation:

**CI/CD Automation:**

Jenkins automates the software development lifecycle, from code integration to deployment, by implementing CI/CD workflows called pipelines.

**Open-Source and Extensible:**

It's an open-source tool written in Java, and its functionality can be extended through a vast ecosystem of plugins.

**Pipeline as Code:**

Jenkins uses "Pipeline as Code" where CI/CD pipelines are defined in a text file (Jenkinsfile) and version-controlled, allowing for better collaboration and repeatability.

**Key Features:**

* Automated Builds and Tests: Jenkins can automatically build, test, and package software, allowing developers to catch errors early.
* Integration with Various Tools: It integrates with various tools like Git, Maven, Gradle, Docker, and more, enabling a streamlined workflow.
* Easy Installation and Configuration: Jenkins is easy to install and configure, with a web-based interface for managing jobs and pipelines.
* Scalability and Reliability: It's designed to handle large-scale CI/CD processes and can be deployed on various platforms.

**Benefits:**

* Faster Software Delivery: Automation leads to faster feedback loops and quicker software releases.
* Improved Code Quality: Early detection of issues through automated testing improves code quality.
* Reduced Manual Effort: Automation frees up developers from repetitive tasks, allowing them to focus on coding and innovation.

1. **Docker.**

AI Overview

Docker is a platform that allows developers to package applications and their dependencies into portable containers, enabling consistent execution across different environments, simplifying development, deployment, and management.

Here's a more detailed explanation:

**What is Docker?**

**Containerization Platform:**

Docker is a containerization platform, meaning it enables developers to package applications and their dependencies into self-contained, portable units called containers.

**Lightweight and Efficient:**

Docker containers are lightweight and share the host system's operating system kernel, leading to faster startup times and efficient resource utilization compared to traditional virtual machines.

**Open Source:**

Docker is an open-source project.

**Docker Engine:**

The core of Docker is the Docker Engine, a client-server application that manages containers.

**Docker Images:**

Docker uses images as templates to create containers. An image contains everything needed to run an application, including code, libraries, and runtime.

**Why Use Docker?**

**Consistency and Portability:**

Docker ensures that applications run consistently across different environments, eliminating the "it works on my machine" problem.

**Simplified Deployment:**

Docker simplifies the deployment process by allowing developers to package and deploy applications as containers, which can be run anywhere.

**Resource Efficiency:**

Docker containers are lightweight and share the host system's kernel, leading to better resource utilization.

**Scalability:**

Docker makes it easier to scale applications by allowing developers to run multiple instances of containers.

**Microservices Architecture:**

Docker is well-suited for building and managing microservices architectures, where applications are broken down into smaller, independent services.

**Continuous Integration and Continuous Deployment (CI/CD):**

Docker facilitates CI/CD pipelines by allowing developers to package applications into containers and deploy them automatically.

**Developer Productivity:**

Docker streamlines the development process by making it easier to build, test, and deploy applications.

**Collaboration:**

Docker enables developers to share and collaborate on containerized applications more easily.