Index: AWS and DEVOPS (ETR: 2 Hrs)

Why AWS for DevOps?

- ✓ AWS provides a set of flexible services designed to enable companies to more rapidly and reliably build and deliver products using AWS and DevOps practices.
- ✓ These services simplify provisioning and managing infrastructure, deploying application code, automating software release processes, and monitoring your application and infrastructure performance.

1. Get Started Fast

- ✓ Each AWS service is ready to use if you have an AWS account.
- ✓ There is no setup required or software to install

2. Fully Managed Services

- ✓ These services can help you take advantage of AWS resources quicker.
- ✓ You can worry less about setting up, installing, and operating infrastructure on your own. This lets you focus on your core product.

3. Built for Scale

- ✓ You can manage a single instance or scale to thousands using AWS services.
- ✓ These services help you make the most of flexible compute resources by simplifying provisioning, configuration, and scaling.

4. Programmable

- ✓ You have the option to use each service via the AWS Command Line Interface or through APIs and SDKs.
- ✓ You can also model and provision AWS resources and your entire AWS infrastructure using declarative AWS CloudFormation templates.

5. Automation

- ✓ AWS helps you use automation so you can build faster and more efficiently.
- ✓ Using AWS services, you can automate manual tasks or processes such as deployments, development & test workflows, container management, and configuration management.

6. Secure

- ✓ Use AWS Identity and Access Management (IAM) to set user permissions and policies.
- ✓ This gives you granular control over who can access your resources and how they access those resources.

7. Large Partner Ecosystem

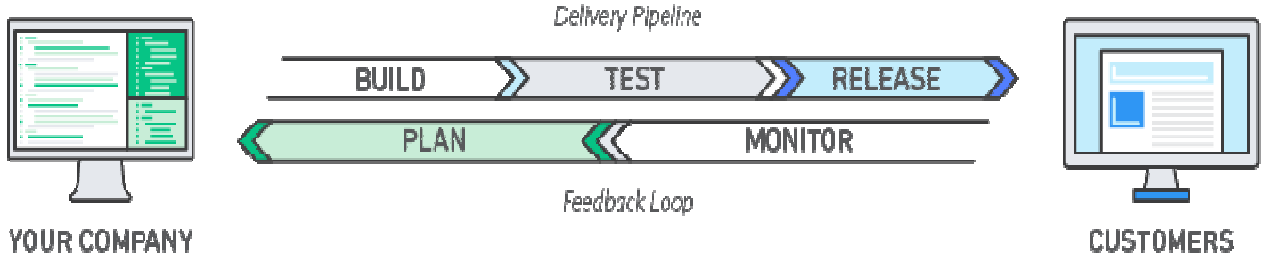
- ✓ AWS supports a large ecosystem of partners which integrate with and extend AWS services.
- ✓ Use your preferred third-party and open source tools with AWS to build an end-to-end solution.

8. Pay-As-You-Go

- ✓ With AWS purchase services as you need them and only for the period when you plan to use them.
- ✓ AWS pricing has no upfront fees, termination penalties, or long term contracts.
- ✓ The AWS Free Tier helps you get started with AWS.

What is DevOps?

- ✓ DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity: evolving and improving products at a faster pace than organizations using traditional software development and infrastructure management processes.
- ✓ This speed enables organizations to better serve their customers and compete more effectively in the market.



How DevOps Works?

- ✓ Under a DevOps model, development and operations teams are no longer “siloed.”
- ✓ Sometimes, these two teams are merged into a single team where the engineers work across the entire application lifecycle, from development and test to deployment to operations, and develop a range of skills not limited to a single function.
- ✓ Quality assurance and security teams may also become more tightly integrated with development and operations and throughout the application lifecycle.
- ✓ These teams use practices to automate processes that historically have been manual and slow.
- ✓ They use a technology stack and tooling which help them operate and evolve applications quickly and reliably.
- ✓ These tools also help engineers independently accomplish tasks (for example, deploying code or provisioning infrastructure) that normally would have required help from other teams, and this further increases a team’s velocity.

Benefits of DevOps

1. Speed

- ✓ Move at high velocity so you can innovate for customers faster, adapt to changing markets better, and grow more efficient at driving business results.
- ✓ The DevOps model enables your developers and operations teams to achieve these results.
- ✓ For example, microservices and continuous delivery let teams take ownership of services and then release updates to them quicker.

2. Rapid Delivery

- ✓ Increase the frequency and pace of releases so you can innovate and improve your product faster.
- ✓ The quicker you can release new features and fix bugs, the faster you can respond to your customers’ needs and build competitive advantage.
- ✓ Continuous integration and continuous deliveryare practices that automate the software release process, from build to deploy.

3. Reliability

- ✓ Ensure the quality of application updates and infrastructure changes so you can reliably deliver at a more rapid pace while maintaining a positive experience for end users.
- ✓ Use practices like continuous integration and continuous delivery to test that each change is functional and safe.
- ✓ Monitoring and logging practices help you stay informed of performance in real-time.

4. Scale

- ✓ Operate and manage your infrastructure and development processes at scale.

- ✓ Automation and consistency help you manage complex or changing systems efficiently and with reduced risk.
- ✓ For example, infrastructure as code helps you manage your development, testing, and production environments in a repeatable and more efficient manner.

5. Improved Collaboration

- ✓ Build more effective teams under a DevOps cultural model, which emphasizes values such as ownership and accountability.
- ✓ Developers and operations teams collaborate closely, share many responsibilities, and combine their workflows.
- ✓ This reduces inefficiencies and saves time (e.g. reduced handover periods between developers and operations, writing code that takes into account the environment in which it is run).

6. Security

- ✓ Move quickly while retaining control and preserving compliance.
- ✓ You can adopt a DevOps model without sacrificing security by using automated compliance policies, fine-grained controls, and configuration management techniques.
- ✓ For example, using infrastructure as code and policy as code, you can define and then track compliance at scale.

Why DevOps Matters?

Software and the Internet have transformed the world and its industries, from shopping to entertainment to banking. Software no longer merely supports a business; rather it becomes an integral component of every part of a business.

- ✓ Companies interact with their customers through software delivered as online services or applications and on all sorts of devices.
- ✓ They also use software to increase operational efficiencies by transforming every part of the value chain, such as logistics, communications, and operations.
- ✓ In a similar way that physical goods companies transformed how they design, build, and deliver products using industrial automation throughout the 20th century, companies in today's world must transform how they build and deliver software.

How to Adopt a DevOps Model?

DevOps Cultural Philosophy:

- ✓ Transitioning to DevOps requires a change in culture and mindset.
- ✓ At its simplest, DevOps is about removing the barriers between two traditionally siloed teams, development and operations.
- ✓ In some organizations, there may not even be separate development and operations teams; engineers may do both.
- ✓ With DevOps, the two teams work together to optimize both the productivity of developers and the reliability of operations.

- ✓ They strive to communicate frequently, increase efficiencies, and improve the quality of services they provide to customers.
- ✓ They take full ownership for their services, often beyond where their stated roles or titles have traditionally been scoped by thinking about the end customer's needs and how they can contribute to solving those needs.
- ✓ Quality assurance and security teams may also become tightly integrated with these teams.
- ✓ Organizations using a DevOps model, regardless of their organizational structure, have teams that view the entire development and infrastructure lifecycle as part of their responsibilities.

DevOps Practices Explained:

- ✓ There are a few key practices that help organizations innovate faster through automating and streamlining the software development and infrastructure management processes. Most of these practices are accomplished with proper tooling.
- ✓ One fundamental practice is to perform very frequent but small updates. This is how organizations innovate faster for their customers.
- ✓ These updates are usually more incremental in nature than the occasional updates performed under traditional release practices.
- ✓ Frequent but small updates make each deployment less risky. They help teams address bugs faster because teams can identify the last deployment that caused the error.
- ✓ Although the cadence and size of updates will vary, organizations using a DevOps model deploy updates much more often than organizations using traditional software development practices.
- ✓ Organizations might also use a microservices architecture to make their applications more flexible and enable quicker innovation.
- ✓ The microservices architecture decouples large, complex systems into simple, independent projects.
- ✓ Applications are broken into many individual components (services) with each service scoped to a single purpose or function and operated independently of its peer services and the application as a whole.
- ✓ This architecture reduces the coordination overhead of updating applications, and when each service is paired with small, agile teams who take ownership of each service, organizations can move more quickly.
- ✓ However, the combination of microservices and increased release frequency leads to significantly more deployments which can present operational challenges.
- ✓ Thus, DevOps practices like continuous integration and continuous delivery solve these issues and let organizations deliver rapidly in a safe and reliable manner.
- ✓ Infrastructure automation practices, like infrastructure as code and configuration management, help to keep computing resources elastic and responsive to frequent changes.
- ✓ In addition, the use of monitoring and logging helps engineers track the performance of applications and infrastructure so they can react quickly to problems.
- ✓ Together, these practices help organizations deliver faster, more reliable updates to their customers.

DevOps Practices

The following are DevOps best practices:

- ✓ Continuous Integration
- ✓ Continuous Delivery
- ✓ Microservices
- ✓ Infrastructure as Code
- ✓ Monitoring and Logging
- ✓ Communication and Collaboration

Best DevOps Practices with AWS:

Continuous Integration

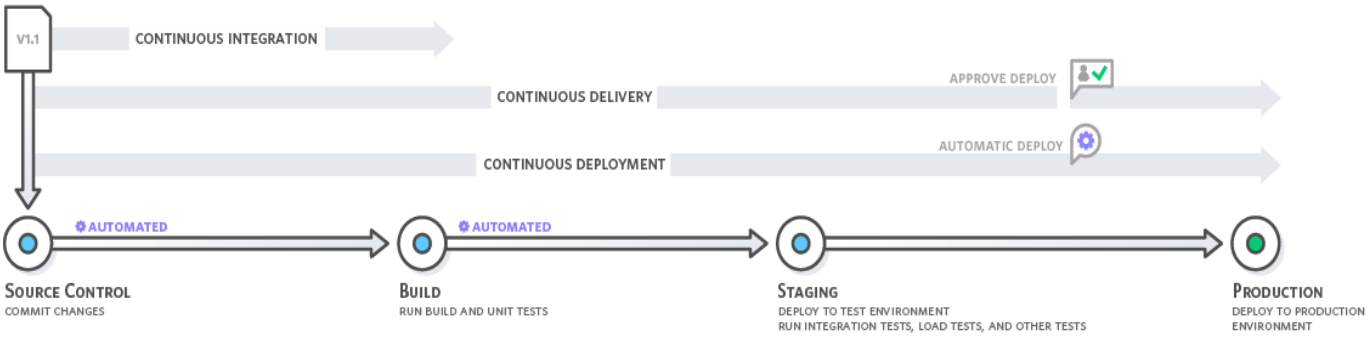
- ✓ Continuous integration is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.
- ✓ Continuous integration most often refers to the build or integration stage of the software release process and entails both an automation component (e.g. a CI or build service) and a cultural component (e.g. learning to integrate frequently).
- ✓ The key goals of continuous integration are to find and address bugs quicker, improve software quality, and reduce the time it takes to validate and release new software updates.

Why is Continuous Integration Needed?

- ✓ In the past, developers on a team might work in isolation for an extended period of time and only attempt to merge their changes to the master branch once their work was completed.
- ✓ This batched process made merging accumulated code changes difficult and time-consuming. This is compounded when small bugs accumulate for a long time without correction. These factors combined made it harder to deliver updates to customers quickly.

How does Continuous Integration Work?

- ✓ With continuous integration, developers frequently commit to a shared repository using a version control system such as Git.
- ✓ Prior to each commit, developers may choose to run local unit tests on their code as an extra verification layer before integrating.
- ✓ A continuous integration service detects commits to the shared repository, and automatically builds and runs unit tests on the new code changes to immediately surface any functional or integration errors.



- ✓ *Continuous integration refers to the build and unit testing stages of the software release process.*
- ✓ *Every revision that is committed triggers an automated build and test.*
- ✓ With continuous delivery, code changes are automatically built, tested, and prepared for a release to production.
- ✓ Continuous delivery expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.

Continuous Integration Benefits

1. Improve Developer Productivity

✓ Continuous integration helps your team be more productive by freeing developers from manual tasks and encouraging behaviors that help reduce the number of errors and bugs released to customers.

2. Find and Address Bugs Quicker

✓ With more frequent testing, your team can discover and address bugs earlier before they grow into larger problems later.

3. Deliver Updates Faster

✓ Continuous integration helps your team deliver updates to their customers faster and more frequently.

TASK:

You can practice continuous integration on AWS in several ways.

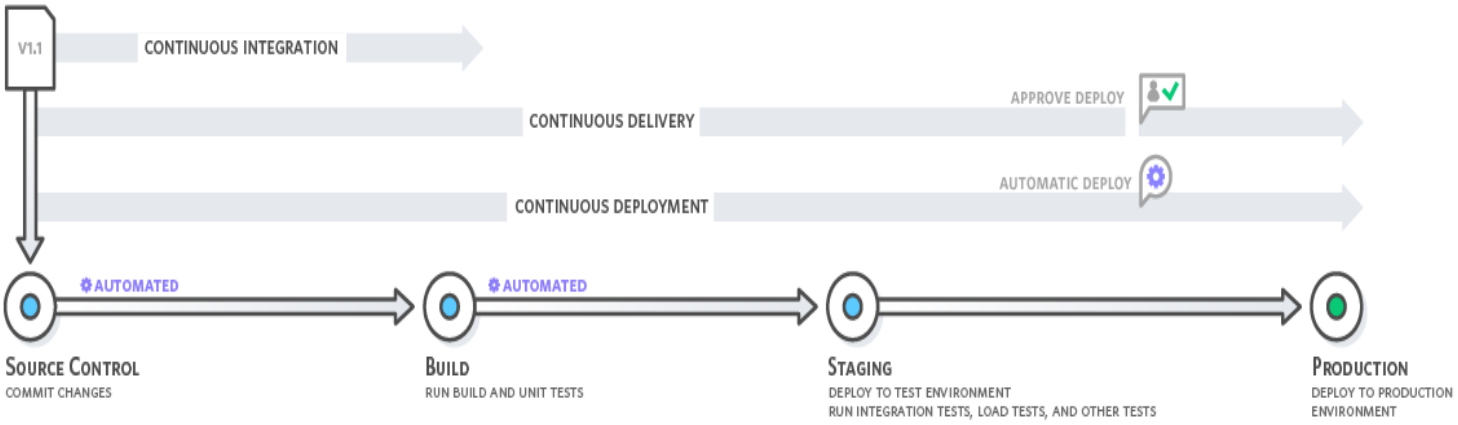
- ✓ Set up a continuous integration workflow with AWS CodePipeline, which lets you build a workflow that builds code in AWS CodeBuild every time you commit a change.

Continuous Delivery

- ✓ Continuous delivery is a software development practice where code changes are automatically built, tested, and prepared for a release to production.
- ✓ It expands upon continuous integration by deploying all code changes to a testing environment and/or a production environment after the build stage.
- ✓ When continuous delivery is implemented properly, developers will always have a deployment-ready build artifact that has passed through a standardized test process.
- ✓ Continuous delivery lets developers automate testing beyond just unit tests so they can verify application updates across multiple dimensions before deploying to customers.
- ✓ These tests may include UI testing, load testing, integration testing, API reliability testing, etc. This helps developers more thoroughly validate updates and pre-emptively discover issues.
- ✓ With the cloud, it is easy and cost-effective to automate the creation and replication of multiple environments for testing, which was previously difficult to do on-premises.

Continuous Delivery vs. Continuous Deployment

- ✓ With continuous delivery, every code change is built, tested, and then pushed to a non-production testing or staging environment.
- ✓ There can be multiple, parallel test stages before a production deployment.
- ✓ The difference between continuous delivery and continuous deployment is the presence of a manual approval to update to production.
- ✓ With continuous deployment, production happens automatically without explicit approval.



- ✓ Continuous delivery automates the entire software release process.
- ✓ Every revision that is committed triggers an automated flow that builds, tests, and then stages the update.
- ✓ The final decision to deploy to a live production environment is triggered by the developer.
- ✓ With continuous deployment, revisions are deployed to a production environment automatically without explicit approval from a developer, making the entire software release process automated.

Continuous Delivery Benefits

1. Automate the Software Release Process

- ✓ Continuous delivery lets your team automatically build, test, and prepare code changes for release to production so that your software delivery is more efficient and rapid

2. Improve Developer Productivity

- ✓ These practices help your team be more productive by freeing developers from manual tasks and encouraging behaviors that help reduce the number of errors and bugs deployed to customers.

3. Find and Address Bugs Quicker

- ✓ Your team can discover and address bugs earlier before they grow into larger problems later with more frequent and comprehensive testing.
- ✓ Continuous delivery lets you more easily perform additional types of tests on your code because the entire process has been automated.

4. Deliver Updates Faster

- ✓ Continuous delivery helps your team deliver updates to customers faster and more frequently.
- ✓ When continuous delivery is implemented properly, you will always have a deployment-ready build artifact that has passed through a standardized test process.

Task:

Practice continuous delivery by using AWS CodePipeline, which lets you build a workflow that builds code in AWS CodeBuild, runs automated tests, and deploys code.

DevOps Tooling by AWS

- ✓ AWS provides services that help you practice DevOps at your company and that are built first for use with AWS.
- ✓ These tools automate manual tasks, help teams manage complex environments at scale, and keep engineers in control of the high velocity that is enabled by DevOps.
- ✓ The DevOps model relies on effective tooling to help teams rapidly and reliably deploy and innovate for their customers.

Continuous Integration and Continuous Delivery with AWS

- ✓ The AWS Developer Tools help you securely store and version your application's source code and automatically build, test, and deploy your application to AWS or your on-premises environment.
- ✓ Start with AWS CodePipeline to build a continuous integration or continuous delivery workflow that uses AWS CodeBuild, AWS CodeDeploy, and other tools, or use each service separately.

Software Release Workflows

1. AWS CodePipeline (alternative Tools- Jenkins,Circle CI,Travis CI,TFS)

- ✓ AWS CodePipeline is a continuous integration and continuous delivery service for fast and reliable application and infrastructure updates.
- ✓ CodePipeline builds, tests, and deploys your code every time there is a code change, based on the release process models you define.
- ✓ This enables you to rapidly and reliably deliver features and updates.

Build and Test Code

2. AWS CodeBuild (Maven, Gradle, Jenkins, Bamboo,Circle CI,Teamcity)

- ✓ AWS CodeBuild is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy.
- ✓ With CodeBuild, you don’t need to provision, manage, and scale your own build servers.
- ✓ CodeBuild scales continuously and processes multiple builds concurrently, so your builds are not left waiting in a queue.

Deployment Automation

3. AWS CodeDeploy (Chef,Ansible,Puppet)

- ✓ AWS CodeDeploy automates code deployments to any instance, including Amazon EC2 instances and on-premises servers.
- ✓ AWS CodeDeploy makes it easier for you to rapidly release new features, helps you avoid downtime during application deployment, and handles the complexity of updating your applications.

Unified CI/CD Projects

4. AWS CodeStar (Heroku, openshift , AWS Elastic Bean stack,Dokku,Azure websites)

- ✓ AWS CodeStar enables you to quickly develop, build, and deploy applications on AWS.
- ✓ AWS CodeStar provides a unified user interface, enabling you to easily manage your software development activities in one place.
- ✓ With AWS CodeStar, you can set up your entire continuous delivery toolchain in minutes, allowing you to start releasing code faster.

Microservices:

- ✓ The microservices architecture is a design approach to build a single application as a set of small services.
- ✓ Each service runs in its own process and communicates with other services through a well-defined interface using a lightweight mechanism, typically an HTTP-based application programming interface (API).
- ✓ Microservices are built around business capabilities; each service is scoped to a single purpose.
- ✓ You can use different frameworks or programming languages to write microservices and deploy them independently, as a single service, or as a group of services.

Microservices with AWS:

Build and deploy a microservices architecture using containers or serverless computing.

Production Docker Platform

5. Amazon Elastic Container Service(Kubernetes, Apache Mesos,Saltstack,Shippable)

- ✓ Amazon Elastic Container Service (ECS) is a highly scalable, high performance container management service that supports Docker containers.
- ✓ It allows you to easily run applications on a managed cluster of Amazon EC2 instances.

Serverless Computing

6. AWS Lambda(Google Cloud Functions, Microsoft Azure Functions , IBM OpenWhisk)

- ✓ AWS Lambda lets you run code without provisioning or managing servers.
- ✓ With Lambda, you can run code for virtually any type of application or backend service - all with zero administration.

- ✓ Just upload your code and Lambda takes care of everything required to run and scale your code with high availability.

Infrastructure as Code

- ✓ Infrastructure as code is a practice in which infrastructure is provisioned and managed using code and software development techniques, such as version control and continuous integration.
- ✓ The cloud's API-driven model enables developers and system administrators to interact with infrastructure programmatically, and at scale, instead of needing to manually set up and configure resources.
- ✓ Thus, engineers can interface with infrastructure using code-based tools and treat infrastructure in a manner similar to how they treat application code.
- ✓ Because they are defined by code, infrastructure and servers can quickly be deployed using standardized patterns, updated with the latest patches and versions, or duplicated in repeatable ways.

Infrastructure as Code with AWS:

Provision, configure, and manage your AWS infrastructure resources using code and templates. Monitor and enforce infrastructure compliance.

Templated Infrastructure Provisioning

7. AWS CloudFormation(Hashicorp Terraform, Redhat CloudForms, VMware vRealize)

- ✓ AWS CloudFormation gives developers and systems administrators an easy way to create and manage a collection of related AWS resources, provisioning and updating them in an orderly and predictable fashion.
- ✓ You can use AWS CloudFormation's sample templates or create your own templates.

Chef Configuration Management

8. **AWS OpsWorks (Ansible, Teamcity, Bamboo,codenvy)**

- ✓ AWS OpsWorks is a configuration management service that uses Chef, an automation platform that treats server configurations as code.
- ✓ OpsWorks uses Chef to automate how servers are configured, deployed, and managed across your Amazon Elastic Compute Cloud (Amazon EC2) instances or on-premises compute environments.
- ✓ OpsWorks has two offerings, AWS Opsworks for Chef Automate, and AWS OpsWorks Stacks.

Configuration Management

- ✓ Developers and system administrators use code to automate operating system and host configuration, operational tasks, and more.
- ✓ The use of code makes configuration changes repeatable and standardized.
- ✓ It frees developers and systems administrators from manually configuring operating systems, system applications, or server software.

Configuration Management with AWS:

9. Amazon EC2 Systems Manager(Freshservice, Microsoft Operations Management Suites, Goverlan Reach, ConnectWise Automate)

- ✓ Amazon EC2 Systems Manager is a management service that helps you automatically collect software inventory, apply OS patches, create system images, and configure Windows and Linux operating systems.
- ✓ These capabilities help you define and track system configurations, prevent drift, and maintain software compliance of your EC2 and on-premises configurations.

Policy as Code

- ✓ With infrastructure and its configuration codified with the cloud, organizations can monitor and enforce compliance dynamically and at scale.
- ✓ Infrastructure that is described by code can thus be tracked, validated, and reconfigured in an automated way.
- ✓ This makes it easier for organizations to govern changes over resources and ensure that security measures are properly enforced in a distributed manner (e.g. information security or compliance with PCI-DSS or HIPAA).
- ✓ This allows teams within an organization to move at higher velocity since non-compliant resources can be automatically flagged for further investigation or even automatically brought back into compliance.

Policy as Code with AWS

10.AWS Config (Ansible, Teamcity, Bamboo,codenvy)

- ✓ AWS Config is a fully managed service that provides you with an AWS resource inventory, configuration history, and configuration change notifications to enable security and governance.
- ✓ Config Rules enables you to create rules that automatically check the configuration of AWS resources recorded by AWS Config.

Monitoring and Logging

- ✓ Organizations monitor metrics and logs to see how application and infrastructure performance impacts the experience of their product's end user.
- ✓ By capturing, categorizing, and then analyzing data and logs generated by applications and infrastructure, organizations understand how changes or updates impact users, shedding insights into the root causes of problems or unexpected changes.
- ✓ Active monitoring becomes increasingly important as services must be available 24/7 and as application and infrastructure update frequency increases.
- ✓ Creating alerts or performing real-time analysis of this data also helps organizations more proactively monitor their services.

Monitoring and Logging with AWS

Record logs and monitor application and infrastructure performance in near real-time.

Cloud and Network Monitoring

10. Amazon CloudWatch(Dynatrace,Datadog,Microsoft System center, NewRelic APM,Splunk,ELK)

- ✓ Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS.
- ✓ You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources.

Distributed Tracing

11.AWS X-Ray (JIRA, Instabug, Bugzilla, Helix ALM,TASKRAY)

- ✓ AWS X-Ray helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture.
- ✓ With X-Ray, you can understand how your application and its underlying services are performing to identify and troubleshoot the root cause of performance issues and errors.

Activity & API Usage Tracking

12.AWS CloudTrail(Anypoint Platform,Dell Boomi,webMethods,Postman)

- ✓ AWS CloudTrail is a web service that records AWS API calls for your account and delivers log files to you.
- ✓ The recorded information includes the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service.

Platform as a Service

Deploy web applications without needing to provision and manage the infrastructure and application stack.

Run and Manage Web Apps

13. AWS Elastic Beanstalk(DigitalOcean, Openshift,Heroku,Azure,GoogleApp Engine)

- ✓ AWS Elastic Beanstalk is an easy-to-use service for deploying and scaling web applications and services developed with Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker on familiar servers such as Apache, Nginx, Passenger, and IIS.
- ✓ You can simply upload your code and Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring.
- ✓ At the same time, you retain full control over the AWS resources powering your application and can access the underlying resources at any time.

Communication and Collaboration

- ✓ Increased communication and collaboration in an organization is one of the key cultural aspects of DevOps.
- ✓ The use of DevOps tooling and automation of the software delivery process establishes collaboration by physically bringing together the workflows and responsibilities of development and operations.
- ✓ Building on top of that, these teams set strong cultural norms around information sharing and facilitating communication through the use of chat applications, issue or project tracking systems, and wikis.
- ✓ This helps speed up communication across developers, operations, and even other teams like marketing or sales, allowing all parts of the organization to align more closely on goals and projects.

Version Control

Host secure, highly scalable Git repositories in the cloud.

Private Git Hosting

14.AWS CodeCommit(Git, GitHub, BitBucket,Mercurial, TFS, SVN, Helix VCS)

- ✓ AWS CodeCommit is a fully-managed source control service that makes it easy for companies to host secure and highly scalable private Git repositories.
- ✓ You can use CodeCommit to securely store anything from source code to binaries, and it works seamlessly with your existing Git tools.
- ✓ Using AWS CodeCommit, Edmunds.com developers have a scalable, highly available source control service that reduces costs and simplifies administration.