**Miscellaneous**

**AWS Resource Access Manager**

* AWS Resource Access Manager (RAM) is a service that enables you to easily and securely share AWS resources with any AWS account or within your AWS Organization. You can share AWS Transit Gateways, Subnets, AWS License Manager configurations, and Amazon Route 53 Resolver rules resources with RAM.
* Many organizations use multiple accounts to create administrative or billing isolation, and to limit the impact of errors as part of the AWS Organizations service.
* RAM eliminates the need to create duplicate resources in multiple accounts, reducing the operational overhead of managing those resources in every single account you own.

**Iot Core**

* AWS IoT Core is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices.
* AWS IoT Core provides secure communication and data processing across different kinds of connected devices and locations so you can easily build IoT applications.

**AWS Workspaces**

* Amazon WorkSpaces is a managed, secure Desktop-as-a-Service (DaaS) solution. You can use Amazon WorkSpaces to provision either Windows or Linux desktops in just a few minutes and quickly scale to provide thousands of desktops to workers across the globe.
* Amazon WorkSpaces helps you eliminate the complexity in managing hardware inventory, OS versions and patches, and Virtual Desktop Infrastructure (VDI), which helps simplify your desktop delivery strategy.
* With Amazon WorkSpaces, your users get a fast, responsive desktop of their choice that they can access anywhere, anytime, from any supported device.

**AWS Fargate**

* AWS Fargate is a serverless compute engine for containers.
* The Fargate launch type allows you to run your containerized applications without the need to provision and manage the backend infrastructure. Just register your task definition and Fargate launches the container for you.
* It works with both Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).

**Blue Green Deployments**

* The Blue-Green deployment approach does this by ensuring you have two production environments, as identical as possible. At any time one of them, let's say blue for the example, is live. As you prepare a new release of your software you do your final stage of testing in the green environment. Once the software is working in the green environment, you switch the router so that all incoming requests go to the green environment - the blue one is now idle.
* Blue-green deployment also gives you a rapid way to rollback - if anything goes wrong you switch the router back to your blue environment.