DAY-II

An Introduction to

amazon web services™

By P.S.R.Patnaik

Microsoft Specialist: Architecting Microsoft Azure Solutions
Microsoft, License F312-3640,
AWS Certified Solutions Architect - Associate
Amazon Web Services, License 639BWPV1JFRE11CR

AWS EFS

AWS EFS
Introduction to File System Storage

Ref. Link

Amazon Elastic File System (Amazon EFS) provides simple, scalable, elastic file storage for use with AWS Cloud services and on-premises resources. It is easy to use and offers a simple interface that allows you to create and configure file systems quickly and easily. Amazon EFS is built to elastically scale on demand without disrupting applications, growing and shrinking automatically as you add and remove files, so your applications have the storage they need, when they need it.

AWS EFS
Understanding Use Cases

Ref. Link

- ENTERPRISE APPLICATIONS
- MEDIA & ENTERTAINMENT WORKFLOWS
- ANALYTICS
- HOME DIRECTORIES
- CONTENT MANAGEMENT & WEB SERVING
- SOFTWARE DEVELOPMENT TOOLS
- DATABASE BACKUPS
- CONTAINER STORAGE

AWS EFS
Introduction to Elastic File System - EFS

Ref. Link

Amazon EC2 instances, enabling your applications to achieve high levels of aggregate throughput and IOPS that scale as a file system grows, with consistent low latencies. Amazon EFS is designed for high availability and durability storing data redundantly across multiple Availability Zones. Amazon EFS is well suited to support a broad spectrum of use cases, including web serving and content management, enterprise applications, media and entertainment processing workflows, home directories, database backups, developer tools, container storage, and big data analytics workloads.

AWS EFS

Configuration of EFS

Ref. Link

Step 1: Create Your EC2 Resources and Launch Your EC2 Instance

Step 2: Create Your Amazon EFS File System

Step 3: Connect to Your Amazon EC2 Instance and Mount the Amazon EFS File System

Step 4: Sync Files from Existing File Systems to Amazon EFS Using EFS File Sync

Step 5: Clean Up Resources and Protect Your AWS Account

AWS EFS

Creating and using EFS with Multiple EC2 Instances

Ref. Link

Create two security groups.

Add rules to the security groups to authorize additional access.

Launch an EC2 instance. You create and mount an Amazon EFS file system on this instance in the next step.

AWS EFS

Troubleshooting in EFS

Ref. Link

Troubleshooting Amazon EFS: General Issues

Troubleshooting File Operation Errors

Troubleshooting AMI and Kernel Issues

Troubleshooting Mount Issues

Troubleshooting Encryption

AWS LB

AWS LB

Introduction to Load Balancer

Ref. Link

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones. Elastic Load Balancing offers three types of load balancers that all feature the high availability, automatic scaling, and robust security necessary to make your applications fault tolerant

AWS LB

Types of Load Balancer in AWS

Ref. Link

1. Application Load Balancer
2. Network Load Balancer
3. Classic Load Balancer

Benefits : Highly available, Secure, Elastic, Flexible, Robust monitoring & auditing, Hybrid load balancing

AWS LB
Use Cases of Load balancers

Ref. Link

- Achieve better fault tolerance for your applications
- Automatically load balance your containerized applications
- Automatically scale your applications
- Using Elastic Load Balancing in your Amazon Virtual Private Cloud (Amazon VPC)
- Hybrid load balancing with Elastic Load Balancing

AWS LB

Important Components of Load Balancer

Ref. Link

- AWS Management Console
- AWS Command Line Interface (AWS CLI)
- AWS SDKs
- Query API

AWS LB

How Health-Check Works for Load Balancer

Ref. Link

You configure active health checks for the targets in a target group using the following settings. The load balancer sends a health check request to each registered target every **Health Check Interval Seconds** seconds, using the specified port, protocol, and ping path. It waits for the target to respond within the response timeout period. If the health checks exceed the threshold for consecutive failed responses, the load balancer takes the target out of service.

AWS LB

Configuration of a load balancer and its management

Ref. Link

Step 1: Select a Load Balancer Type
Step 2: Define Your Load Balancer
Step 3: Assign Security Groups to Your Load Balancer in a VPC
Step 4: Configure Health Checks for Your EC2 Instances
Step 5: Register EC2 Instances with Your Load Balancer
Step 6: Tag Your Load Balancer (Optional)
Step 7: Create and Verify Your Load Balancer
Step 8: Delete Your Load Balancer (Optional)

AWS LB
Pricing - Load balancer

Ref. Link

Application Load Balancer

You are charged for each hour or partial hour that an Application Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used per hour.

Network Load Balancer

You are charged for each hour or partial hour that a Network Load Balancer is running and the number of Load Balancer Capacity Units (LCU) used by Network Load Balancer per hour.

Classic Load Balancer

You are charged for each hour or partial hour that a Classic Load Balancer is running and for each GB of data transferred through your load balancer.

AWS LB

Limitations and Best Practice of Using Load Balancer

Ref. Link

There are several ways to get started with the Elastic Load Balancing. You can set up an Application Load Balancer or Network Load Balancer with APIs, AWS Command Line Interfaces (CLI), or through the AWS Management Console.

When you use Elastic Load Balancing with Auto Scaling, you can build highly available, fault-tolerant applications which automatically scale capacity up or down based on fluctuations in demand.

AWS AS

AWS AS
Introduction to AutoScaling

Ref. Link

AWS Auto Scaling monitors your applications and automatically adjusts capacity to maintain steady, predictable performance at the lowest possible cost. Using AWS Auto Scaling, you can setup scaling for multiple resources across multiple services in minutes. AWS Auto Scaling provides a simple, powerful user interface that lets you build scaling plans for Amazon EC2 instances and Spot Fleets, Amazon ECS tasks, Amazon DynamoDB tables, and Amazon Aurora Replicas.

AWS AS
Use Cases of Autoscaling

Ref. Link

Adding Amazon EC2 Auto Scaling to your application architecture is one way to maximize the benefits of the AWS cloud. When you use Amazon EC2 Auto Scaling, your applications gain the following benefits:

- Better fault tolerance.
- Better availability.
- Better cost management.

AWS AS

Important Components of AutoScaling

Ref. Link

Groups : Your EC2 instances are organized in to *groups* so that they can be treated as a logical unit for the purposes of scaling and management. When you create a group, you can specify its minimum, maximum, and, desired number of EC2 instances. For more information, see Auto Scaling Groups.

Launch configurations : Your group uses a *launch configuration* as a template for its EC2 instances. When you create a launch configuration, you can specify information such as the AMI ID, instance type, key pair, security groups, and block device mapping for your instances. For more information, see Launch Configurations.

Scaling options : Amazon EC2 Auto Scaling provides several ways for you to scale your Auto Scaling groups. For example, you can configure a group to scale based on the occurrence of specified conditions (dynamic scaling) or on a schedule. For more information, see Scaling Options.

AWS AS
Configuration of Autoscaling

Ref. Link

A launch configuration is an instance configuration template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, one or more security groups, and a block device mapping. If you've launched an EC2 instance before, you specified the same information in order to launch the instance.

AWS AS

Launch Configuration and LB Group

Ref. Link

A launch template is similar to a launch configuration, in that it specifies instance configuration information such as the ID of the Amazon Machine Image (AMI), the instance type, a key pair, security groups, and other parameters that you use to launch EC2 instances.

You can attach a load balancer to your Auto Scaling group. The load balancer automatically distributes incoming traffic across the instances in the group.

AWS AS
Pricing

Ref. Link

There is no additional charge for AWS Auto Scaling. You pay only for the AWS resources needed to run your applications and Amazon CloudWatch monitoring fees.

Amazon CloudWatch pricing

Amazon EC2 pricing

Amazon EC2 Spot Fleet pricing

Amazon Elastic Container Service (ECS) pricing

Amazon DynamoDB pricing

Amazon Aurora pricing

AWS AS

Limitation and Best Practice of AutoScaling

Ref. Link

AWS Trusted Advisor provides best practices (or checks) in five categories: cost optimization, security, fault tolerance, performance, and service limits. The status of the check is shown by using color coding on the dashboard page:

Red: action recommended

Yellow: investigation recommended

Green: no problem detected

AWS VPC

AWS VPC

Basics of Networking

Ref. Link

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

Networking Components

- Network Interfaces
- Route Tables
- Internet Gateways
- Egress-Only Internet Gateways
- DHCP Options Sets
- DNS
- Elastic IP Addresses
- VPC Endpoints
- NAT
- VPC Peering
- ClassicLink

AWS VPC
IP Address and CIDR Block

Ref. Link

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16. This is the primary CIDR block for your VPC.

AWS VPC
Concept of Virtual Cloud - Private

Ref. Link

Amazon VPC is the networking layer for Amazon EC2.

- VPCs and Subnets
- Supported Platforms
- Default and Nondefault VPCs
- Accessing the Internet
- Accessing a Corporate or Home Network
- Accessing Services Through AWS PrivateLink

AWS VPC

Introduction to Virtual Private Cloud -VPC

Ref. Link

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways. You can use both IPv4 and IPv6 in your VPC for secure and easy access to resources and applications.

AWS VPC
Subnet and Route Tables

Ref. Link

A route table contains a set of rules, called routes, that are used to determine where network traffic is directed.

Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

- Route Table Basics

AWS VPC
Internet Gateway and NAT

Ref. Link

An internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.

You can use a network address translation (NAT) gateway to enable instances in a private subnet to connect to the internet or other AWS services, but prevent the internet from initiating a connection with those instances.

AWS VPC

Creating and managing a NAT Instance

Ref. Link

You can use the VPC wizard to set up a VPC with a NAT instance; for more information, see Scenario 2: VPC with Public and Private Subnets (NAT). The wizard performs many of the configuration steps for you, including launching a NAT instance, and setting up the routing. However, if you prefer, you can create and configure a VPC and a NAT instance manually.

AWS VPC
Network Interface - NI

Ref. Link

An elastic network interface is a virtual network interface that can include the following attributes:

- a primary private IPv4 address
- one or more secondary private IPv4 addresses
- one Elastic IP address per private IPv4 address
- one public IPv4 address, which can be auto-assigned to the network interface for eth0 when you launch an instance
- one or more IPv6 addresses
- one or more security groups
- a MAC address
- a source/destination check flag
- a description

AWS VPC
Access Control List - ACL

Ref. Link

A network access control list (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC.

AWS VPC
VPC Peering

Ref. Link

A VPC peering connection is a networking connection between two VPCs that enables you to route traffic between them privately. Instances in either VPC can communicate with each other as if they are within the same network. You can create a VPC peering connection between your own VPCs, with a VPC in another AWS account, or with a VPC in a different AWS Region.

AWS uses the existing infrastructure of a VPC to create a VPC peering connection; it is neither a gateway nor a VPN connection, and does not rely on a separate piece of physical hardware.

AWS VPC
Endpoints

Ref. Link

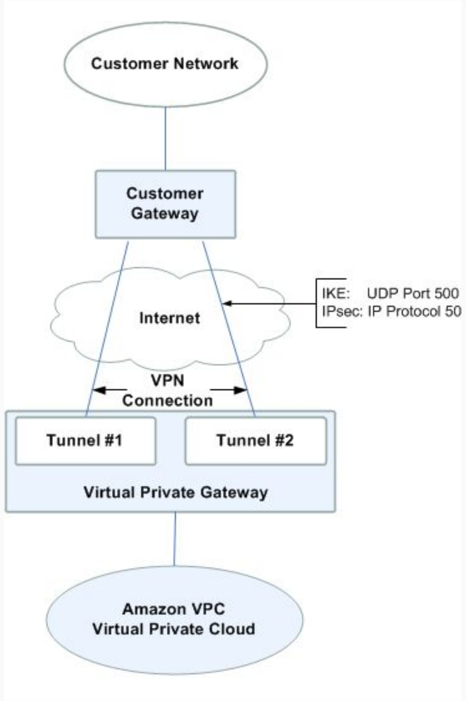
A VPC endpoint enables you to privately connect your VPC to supported AWS services and VPC endpoint services powered by PrivateLink without requiring an internet gateway, NAT device, VPN connection, or AWS Direct Connect connection.

Endpoints are virtual devices. They are horizontally scaled, redundant, and highly available VPC components that allow communication between instances in your VPC and services without imposing availability risks or bandwidth constraints on your network traffic.

AWS VPC
VPN and CGW

Ref. Link

An Amazon VPC VPN connection links your data center (or network) to your Amazon VPC virtual private cloud (VPC). A customer gateway is the anchor on your side of that connection. It can be a physical or software appliance. The anchor on the AWS side of the VPN connection is called a virtual private gateway.



AWS Route 53

AWS Route 53
Introduction to DNS

Ref. Link

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service. It is designed to give developers and businesses an extremely reliable and cost effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.

AWS Route 53
Introduction to Route53

Ref. Link

Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets – and can also be used to route users to infrastructure outside of AWS. You can use Amazon Route 53 to configure DNS health checks to route traffic to healthy endpoints or to independently monitor the health of your application and its endpoints. Amazon Route 53 Traffic Flow makes it easy for you to manage traffic globally.

AWS Route 53

How Route53 Works

Ref. Link

Amazon Route 53 is a highly available and scalable Domain Name System (DNS) web service. You can use Route 53 to perform three main functions in any combination: domain registration, DNS routing, and health checking.

AWS Route 53
Domain Registration in Route53

Ref. Link

When you want to get a new domain name, such as the example.com part of the URL http://example.com, you can register it with Amazon Route 53. You can also transfer the registration for existing domains from other registrars to Route 53 or transfer the registration for domains that you register with Route 53 to another registrar.

The procedures in this chapter explain how to register and transfer domains using the Route 53 console, and how to edit domain settings and view domain status.

AWS Route 53
Health Checks in Route53

Ref. Link

Amazon Route 53 health checks monitor the health and performance of your web applications, web servers, and other resources. Each health check that you create can monitor one of the following:

- The health of a specified resource, such as a web server
- The status of other health checks
- The status of an Amazon CloudWatch alarm

AWS Route 53

Routing Policies in Route53

Ref. Link

When you create a record, you choose a routing policy, which determines how Amazon Route 53 responds to queries:

- Simple routing policy
- Failover routing policy
- Geolocation routing policy
- Geoproximity routing policy
- Latency routing policy
- Multivalue answer routing policy
- Weighted routing policy

AWS Route 53
Creating and Managing different Routing Policies

Ref. Link

After you create a hosted zone for your domain, such as example.com, you create records to tell the Domain Name System (DNS) how you want traffic to be routed for that domain.

For example, you might create records that cause DNS to do the following:

- Route internet traffic for example.com to the IP address of a host in your data center.
- Route email for that domain (ichiro@example.com) to a mail server (mail.example.com).
- Route traffic for a subdomain called operations.tokyo.example.com to the IP address of a different host.

AWS Route 53

Records Sets supported by Route53

Ref. Link

Amazon Route 53 supports the DNS record types.

- A Record Type
- AAAA Record Type
- CAA Record Type
- CNAME Record Type
- MX Record Type
- NAPTR Record Type
- NS Record Type
- PTR Record Type
- SOA Record Type
- SPF Record Type
- SRV Record Type
- TXT Record Type

AWS Route 53

Alarms and Notifications in Route53

Ref. Link

You monitor the status of your health checks on the Amazon Route 53 console. You can also set CloudWatch alarms and get automated notifications when the status of your health check status changes.

- Viewing Health Check Status and the Reason for Health Check Failures
- Monitoring the Latency Between Health Checkers and Your Endpoint
- Monitoring Health Checks Using CloudWatch

AWS Route 53

Limitations & Best Practices in Route53

Enabling automatic renewal for your domains registered with AWS or transferred to AWS will guarantee you full control over domain names registration. When your domains are automatically renewed before their expiration date, the risk of losing them is practically zero.

AWS Route 53
Pricing - Route53

Ref. Link

Pay only for what you use. There is no minimum fee. Estimate your monthly bill using the AWS Simple Monthly Calculator.

Hosted Zones
$0.50 per hosted zone / month for the first 25 hosted zones

$0.10 per hosted zone / month for additional hosted zones

The monthly hosted zone prices listed above are not prorated for partial months. A hosted zone is charged at the time it's created and on the first day of each subsequent month. To allow testing, a hosted zone that is deleted within 12 hours of creation is not charged, however, any queries on that hosted zone will still incur charges

AWS Cloudwatch

AWS Cloudwatch

Introduction to CloudWatch

Ref. Link

Amazon CloudWatch is a monitoring and management service built for developers, system operators, site reliability engineers (SRE), and IT managers. CloudWatch provides you with data and actionable insights to monitor your applications, understand and respond to system-wide performance changes, optimize resource utilization, and get a unified view of operational health. CloudWatch collects monitoring and operational data in the form of logs, metrics, and events, providing you with a unified view of AWS resources, applications and services that run on AWS, and on-premises servers.

AWS Cloudwatch

Important Components of CloudWatch

Ref. Link

You can access CloudWatch using any of the following methods:

Amazon CloudWatch console https://console.aws.amazon.com/cloudwatch/

AWS CLI — For more information, see Getting Set Up with the AWS Command Line Interface in the AWS Command Line Interface User Guide.

CloudWatch API — For more information, see the Amazon CloudWatch API Reference.

AWS SDKs — For more information, see Tools for Amazon Web Services.

AWS Cloudwatch

Creating and Managing metrics in CloudWatch

Ref. Link

- Log in to the instance through the AWS CLI.
- Copy the following Bash script, and then save it to your instance.
- After creating the Bash script, give execute permissions to the file.
- Run the Bash script to check that it works.

AWS Cloudwatch

Creating and Managing Events in CloudWatch

Ref. Link

You can configure the following AWS services as targets for CloudWatch Events:

- Amazon EC2 instances
- AWS Lambda functions
- Amazon Kinesis Data Streams
- Delivery - Kinesis Data Firehose
- Amazon ECS tasks
- Systems Manager Run Command
- Systems Manager Automation
- AWS Batch jobs
- Step Functions state machines
- Pipelines in AWS CodePipeline
- AWS CodeBuild projects
- Amazon Inspector assessment templates
- Amazon SNS topics
- Amazon SQS queues
- Built-in targets—EC2 CreateSnapshot API call, EC2 RebootInstances API call, EC2 StopInstances API call, and EC2 TerminateInstances API call.
- The default event bus of another AWS account

AWS Cloudwatch

Creating and Managing Dashboards in Cloudwatch

Ref. Link

Amazon CloudWatch dashboards are customizable home pages in the CloudWatch console that you can use to monitor your resources in a single view, even those resources that are spread across different Regions. You can use CloudWatch dashboards to create customized views of the metrics and alarms for your AWS resources.

AWS Cloudwatch

Creating and Managing Alarms in CloudWatch

Ref. Link

You can choose specific metrics to trigger the alarm and specify thresholds for those metrics. You can then set your alarm to change state when a metric exceeds a threshold that you have defined.

1. Open the CloudWatch console at https://console.aws.amazon.com/cloudwatch/.
2. In the navigation pane, choose Alarms, Create Alarm.
3. Select Metric step
4. Define Alarm step

AWS Cloudwatch
Logs and Log agent in Cloudwatch

Ref. Link

To collect logs from your Amazon EC2 instances and on-premises servers into CloudWatch Logs, AWS offers both a new unified CloudWatch agent, and an older CloudWatch Logs agent. We recommend the unified CloudWatch agent.

- You can collect both logs and advanced metrics with the installation and configuration of just one agent.
- The unified agent enables the collection of logs from servers running Windows Server.
- If you are using the agent to collect CloudWatch metrics, the unified agent also enables the collection of additional system metrics, for in-guest visibility.

AWS Cloudwatch

Pricing in CloudWatch

Ref. Link

You can get started with Amazon CloudWatch for free. Most AWS Services (EC2, S3, Kinesis, etc.) vend metrics automatically for free to CloudWatch. Many applications should be able to operate within these free tier limits. You can learn more about AWS Free Tier here.

AWS Cloudwatch

Limitations and Best Practices - CloudWatch

Ref. Link

Make monitoring a priority to head off small problems before they become big ones.

Create and implement a monitoring plan that collects monitoring data from all of the parts in your AWS solution so that you can more easily debug a multi-point failure if one occurs.

Automate monitoring tasks as much as possible.

Check the log files on your EC2 instances.

DAY-II
THE END

Ref. Link

END