Kick Off Your Digital Transformation with these 10 Best Practices for AWS Design

TOC

Introduction

Your organization has probably already started to venture out into the world beyond your private data center, but, if it hasn't, then somewhere in your organization, someone is almost certainly considering it.

When moving your workloads to a cloud provider, you should be aware that there are fundamental differences between how you are accustomed to working today and how you will be consuming resources from your cloud provider going forward. According to Gartner's Magic Quadrant for Public Cloud Infrastructure Managed Service Providers, AWS is still the leading service provider when it comes to the public cloud. This white paper will be focusing exclusively on best practices for AWS, although some of these practices can be used in any public cloud provider.

Best Practices for Your AWS Migration

Once you have decided to migrate a solution to or implement a new product in AWS, you can plan ahead to optimize your efficiency and ease your future pain. The top ten ways to prepare for your AWS migration are listed below.

1. VPC Networks and Network Segmentation

Networks are a fundamental part of your design when you move to the cloud.

A virtual private network allows you to carve out a small piece of the cloud for your own private usage. This piece can be located anywhere in AWS. To implement a virtual private network, you need to plan your network topology correctly. While it is easy to use the default VPC (172.31.0.0/16) that Amazon provides, it is unlikely to be valid for your specific case. Chances are you already have an existing network topology which could span a single location or many locations around the globe. Invest the time into understanding and planning your expansion to the cloud to make sure the address spaces do not overlap. Re-deploying all the instances in your VPC's because of an IP range conflict is painful; a little preparation by your current network engineers can avoid this situation.

When starting out, provision your VPC with as big of a subnet range as possible. The largest range that AWS allows is a /16 block of addresses. A big subnet range will maintain your flexibility while also allowing your business to grow.

In a VPC, there are two constructs which you can use to protect network flow into and out of your instances. One is NACLs (Network Access Control Lists) and the other is security groups.

NACLs are usually (although not exclusively) associated with subnets. Security groups are associated with Elastic Network Interfaces (ENIs) or instances.

There is a big difference between the two: NACLs are stateless, and security groups are stateful. A full comparison of these can be found in the AWS documentation. When starting out, it is best to use only security groups. Add NACLs only if a specific requirement mandates them.

2. IAM and Access to Resources

Because of the public nature of the cloud, you should never store credentials in the clear when using AWS access keys. The widespread accessibility of access keys makes them even more powerful than regular passwords. Compromised credentials allow your account to be used anywhere, potentially posing a serious security issue when employees leave the company and you can no longer control their cloud access. Encrypt credentials and store them in a safe place. There are a number of open-source projects available such as AWS-Vault and, if you are a Hashicorp user, Vault.

When using credentials within the AWS cloud, you should be using only IAM roles and not storing credentials inside the instances themselves. Your roles should be as specific as possible; i.e., do not assign wildcard policies to roles, such as a role that will allow access to all EC2 actions.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1528913870335",
      "Action": "ec2:*",
      "Effect": "Allow",
      "Resource": "*"
    }
  ]
}
```

Make a standard naming convention for both the roles and the policies assigned to these roles a standard operational practice. You can use open-source projects such as Repokid or Cloudtracker to help you define the exact scope of permissions that your IAM role requires.

When assigning roles to different people in your organization, combine them into groups. By assigning permissions (policies) to the groups and not to the individuals, you will make overall user management easier.

Make sure to control the number of people and accounts that can actually create new users and access keys in the account.

3. Federated Access

Central management of your users is critical. While IAM is a great place to manage your AWS users, you should look into the option of implementing federation of the user management between your current user directory (which is usually a Microsoft Active Directory) and AWS. There are a number of third party providers that can help enable you to achieve this, such as Auth0, OneLogin, Ping Identity, and more (see the full list of providers here). Federation is done with SAML 2.0 that your Identity provider passes to IAM and with the token your user receives. The user can manage resources inside AWS.

As you continue your day-to-day practice of user management, you can assign roles to specific groups in your organization without having to duplicate the company's user structure in AWS. Not only does this practice provide you with more control, it ensures that all activity in your AWS account will only occur from within your security boundaries by people that have access to your organization's network, either on-premises or through a VPN connection.

4. Design for Failure

It is commonly thought that cloud services provide immediate high availability. This is not the case. Cloud allows you to plan, design, and provision resources is in a very simple way, and it provides you with the tools to achieve a higher level of availability and redundancy than the one that can be achieved when deploying your workloads on-premises. However, this does not mean it comes ready to use "out of the box." Fortunately, there are a lot of tools in AWS that can help you achieve your goals. AWS auto scaling and elastic load balancers are just a couple of the tools that can help you. In addition, database redundancy and availability are built into the DBaaS offerings in AWS. RDS or DynamoDB do all the heavy lifting for you by managing the replication of your data and redundancy in the background. This allows you to invest your time and resources into your applications and solutions. Proper planning and design of your applications is required to make use of all the options AWS offers.

Although it may seem completely reliable, the AWS platform is not without its problems or outages. There is a very specific SLA that AWS provides for their services. However, its availability is not as high as you might like. For some of the services, you should expect only three nines, and for others, eleven nines (S3). This means that, in all practicality, you will never lose a single bit of data once it has been written to S3. It is not a question of if there will be a failure in the cloud, but rather of **when** there will be one. You must design your applications and systems to survive a failure. There are a few concepts in the AWS infrastructure that you should take into account as part of this design process.

Manage your risk and exposure to failure by always deploying into more than one availability zone. Availability zones (AZs) are different physical data centers within a single geographical region that are connected by a high speed link. Using multiple AZs allows you to ensure that you are protected from one data center failure. To protect yourself against a failure in an entire region (i.e., all the availability zones are down or impaired), you will need to make use of the different regions AWS offers around the globe. You should be aware that managing a globally distributed application is not for the faint of heart; a solid architecture is required to support it.

In addition to understanding the concept of availability of compute resources, you should also understand the options available for protecting data between AZs and regions. S3 replication, snapshots, and database replication are available. All you need to do is design your applications properly to make use of these features.

5. Understanding Your Workloads

The latest Gartner Magic Quadrant stated that most lift and shift cloud projects will not succeed. You cannot simply lift what you have inside one location's data center and shift it to another location (in this case, AWS) in exactly the same way using the same architecture as before. The infrastructure that you have built on-premises will not be the same as in AWS, and, therefore, the assumptions that were originally made for your applications are not likely to hold up when deploying in the cloud. There will be a high rate of failure because you are not using the full functionality of the underlying infrastructure.

Assess what you currently have inside your data center. Without knowing what kind of workloads you are running, what is your CPU/RAM/disk/network usage? Will you need specific kinds of instances, ones with more RAM, faster CPU's, or perhaps GPU's? Answering these questions will enable you to make educated decisions about what you should be using in AWS and how to leverage its many features.

You can make use of Cloudwatch and its built-in metrics to help you with this task. Be aware that at the standard level, there are only a limited number of metrics available. It is important to know that there are no metrics available for RAM or disk I/O by default. You can implement your own by following the AWS instructions here.

6. Prepare for a Hybrid Solution

Presumably, you have already made the decision to move to AWS. In order to make this move, you will need connectivity between your on-premises data center and AWS.

Plan and create a solid strategy for a hybrid solution in the interim. You will use this until you have moved everything from your data center to the cloud. There are very few companies that move one hundred percent of all their data center applications into AWS, although there are extraordinary cases, such as Netflix, that have done so. Most enterprises have baggage and

legacy applications which require an interim plan for that period of time when you will need to run a portion of your applications in the cloud and a portion on-premises. VPNs, direct connect, and, once more, proper network design are required to complete the full picture.

Make use of the Virtual Private Gateway service that AWS provides. It is an easy-to-use solution which supports almost all of the common firewall solutions on the market today. It is important to note that the VPN does have its limitations. For example, there is no NAT capability on the AWS VPN side which could prevent proper network connectivity in some scenarios.

7. Automation

Depending on the level of expertise you have in-house, you may be accustomed to deploying applications, servers, networks, or storage in certain ways—ways which may or may not be automated. One of the biggest advantages of using a cloud provider is the fact that everything is an API call. To do anything in the cloud, you only need to issue a command through the API to automate, scale, repeat, and provision any available resource in AWS. Manually provisioning resources through the console (UI) is useful if you want to understand how something works; however, if you find yourself repeating a process or procedure more than once, it should be automated. Automation will enable you to manage infrastructure as code in a secure, repeatable, and optimized manner.

Many of the modern infrastructure tools such as Terraform, Ansible, and Cloudformation will help you on your way towards greater automation.

CloudFormation is usually the first to support a new feature that is released by AWS. It often takes the other open-source projects longer to add support for bleeding edge features, making it necessary to factor this additional delay into your workflow. Each of these tools have their advantages and disadvantages as well. The most popular tool today in the market is Terraform, due to its ease of use and broad support for all cloud vendors and other third party modules.

8. Auditing

Your local regulatory body requires that you track all of your users' actions in the cloud. To turn on the tracking that is built into the AWS platform, simply enable CloudTrail logs on your account across all of the regions. This will make sure that all API calls and all actions that were performed in your account over time are tracked. The collected logs should be sent to an external location—ideally, into a completely separate AWS account and into an S3 bucket where the data can be analyzed and monitored for anomalies. This will ensure that the paper trail cannot be tampered with and will make your INFOSEC people very happy.

The process for this is as follows:

1. Create an IAM role in each of the child accounts that you want to collect files from.

2. For each of these IAM roles, create an access policy that grants them the rights to store the CloudTrail logs in an S3 bucket that is owned by the shared INFOSEC account. You can find full details about this process and implementation here
3. Implement a mechanism that will ship the logs from S3 into ElasticSearch. This will allow you to manage and monitor the activity of your organization at scale.

If tracking all traffic inside your VPCs is a compliance requirement, then you should enable VPC flow logs to trace all the incoming and outgoing traffic to and from your VPC. Be advised that there is an additional cost to use these flow logs. The price will vary based on your usage.

9. Cost and Optimization

For the acquisition of new IT services, your current procurement process is well defined. AWS provides a list of different consumable resources, different instance types, storage tiers detailed, or standard metrics. You will probably not choose the right resources from day one. Typically, the workloads you currently have will not match the corresponding flavors in AWS. Find something which is close enough to your current workloads, run the application for some time, and, during that period, monitor the application. You will be able to recognize if the resource is either overutilized or underutilized. Then, instead of purchasing a whole new piece of equipment to handle the proper load, just resize the workload in AWS. Continuously monitor your resources for proper optimization to identify where you can save on costs.

Invest the time into understanding where reserved instances can save money and when your applications are built correctly. Spot instances will create additional savings above and beyond those.

10. Making Use of Containers in AWS

There are currently three different ways that you can use containers within your AWS account, each with a different use case:

- ECS
- EKS
- Fargate

ECS allows you to provision native docker containers on AWS. When using ECS, you do not have to worry about management of resources. You can use the methodologies you use in-house today. EKS is the latest offering which allows you to offload the management of your Kubernetes cluster to AWS and just worry about your workloads. This does require some management and operational expertise on your part.

The real game changer will be Fargate, which currently only supports ECS. EKS support is planned for later in 2018. Fargate will truly allow you to forget about heavy lifting of managing

container workloads and enable you to specify only the workloads you want to run. It will then schedule them on either of the solutions for you, preventing you from having to worry about the management of the computer resources underneath.

Summary

This white paper has touched upon ten different best practices that you should understand and follow when planning any AWS migration project. In addition to these, there are many other services and design considerations that you will need to take into account. The more you use AWS and their services, the more familiar you will become with these features.

AWS release updates to their products at an astounding pace. There were over 1400 new updates in 2017, including new services and products. The cloud moves at a whole different speed that what you are probably accustomed to, and, at first, this pace will seem overwhelming. Nevertheless, you must still do your research and properly plan for your migration to the cloud.

Fifty percent of cloud migration projects end up over budget or are not completed within the original planned time frame. There are many reasons for this, including a lack of organization, an insufficient budget, and not enough upfront planning. Engage with AWS at an early stage in your project. With proper planning and the appropriate support, you can avoid falling into that statistic of unsuccessful projects by designing a successful AWS cloud migration that sets up your company for greater security.