# Getting Started: Best Practices for Deploying New Environments in the Cloud for the First Time – Part 1




July 16, 2020 Best Practices for Building First-Time Cloud Deployment Audrey Gerber

When organizations take their first steps to use public cloud services, they tend to look at a specific target.

My recommendation – think scale!

Plan a couple of steps ahead instead of looking at a single server that serves just a few customers. Think about a large environment comprised of hundreds or thousands of servers, serving 10,000 customers concurrently.

Planning will allow you to manage the environment (infrastructure, information security and budget) when you do reach a scale of thousands of concurrent customers. The more we plan the deployment of new environments in advance, according to their business purposes and required resources required for each environment, it will be easier to plan to scale up, while maintaining high level security, budget and change management control and more.

In this three-part blog series, we will review some of the most important topics that will help avoid mistakes while building new cloud environments for the first time.

## Resource allocation planning

The first step in resources allocation planning is to decide how to divide resources based on an organizational structure (sales, HR, infrastructure, etc.) or based on environments (production, Dev, testing, etc.)

In-order to avoid mixing resources (or access rights) between various environments, the best practice is to separate the environments as follows:

- Share resource account (security products, auditing, billing management, etc.)
- Development environment account (consider creating separate account for test environment purposes)
- Production environment account

Separating different accounts or environments can be done using:

- Azure Subscriptions or Azure Resource Groups
- AWS Accounts
- GCP Projects
- Oracle Cloud Infrastructure Compartments

## Tagging resources

Even when deploying a single server inside a network environment (AWS VPC, Azure Resource Group, GCP VPC), it is important to tag resources. This allows identifying which resources belong to which projects / departments / environments, for billing purposes.

Common tagging examples:

- Project
- Department
- Environment (Prod, Dev, Test)

Beyond tagging, it is recommended to add a description to resources that support this kind of meta-data, in-order to locate resources by their target use.

## Authentication, Authorization and Password Policy

In order to ease the management of working with accounts in the cloud (and in the future, multiple accounts according to the various environments), the best practice is to follow the rules below:

- Central authentication – In case the organization isn't using Active Directory for central account management and access rights, the alternative is to use managed services such as AWS IAM, Google Cloud IAM, Azure AD, Oracle Cloud IAM, etc.

If managed IAM service is chosen, it is critical to set password policy according to the organization's password policy (minimum password length, password complexity, password history, etc.)

- If the central directory service is used by the organization, it is recommended to connect and sync the managed IAM service in the cloud to the organizational center directory service on premise (federated authentication).
- It is crucial to protect privileged accounts in the cloud environment (such as AWS Root Account, Azure Global Admin, Azure Subscription Owner, GCP Project Owner, Oracle Cloud Service Administrator, etc.), among others, by limiting the use of privileged accounts to the minimum required, enforcing complex passwords, and password rotation every few months. This enables multi-factor authentication and auditing on privileged accounts, etc.
- Access to resources should be defined according to the least privilege principle.
- Access to resources should be set to groups instead of specific users.
- Access to resources should be based on roles in AWS, Azure, GCP, Oracle Cloud, etc.

## Audit Trail

It is important to enable auditing in all cloud environments, in-order to gain insights on access to resources, actions performed in the cloud environment and by whom. This is both security and change management reasons.

Common managed audit trail services:

- AWS CloudTrail – It is recommended to enable auditing on all regions and forward the audit logs to a central S3 bucket in a central AWS account (which will be accessible only for a limited amount of user accounts).
- Working with Azure, it is recommended to enable the use of Azure Monitor for the first phase, in-order to audit all access to resources and actions done inside the subscription. Later on, when the environment expands, you may consider using services such as Azure Security Center and Azure Sentinel for auditing purposes.
- Google Cloud Logging – It is recommended to enable auditing on all GCP projects and forward the audit logs to the central GCP project (which will be accessible only for a limited amount of user accounts).
- Oracle Cloud Infrastructure Audit service – It is recommended to enable auditing on all compartments and forward the audit logs to the Root compartment account (which will be accessible only for a limited amount of user accounts).

**About the author**

Eyal Estrin is a cloud architect, working in the Inter-University Computation Center in Israel (IUCC). He has more than 20 years of experience in infrastructure, information security and public cloud services. He is a public columnist and shares knowledge about cloud services. You can follow him on Twitter at @eyalestrin

# Getting Started: Best Practices for Deploying New Environments in the Cloud for the First Time – Part 2




July 16, 2020 Best Practices for Building First-Time Cloud Deployment Audrey Gerber

In Part 1 of this three-part blog series, we reviewed Resource Allocation Planning, Tagging Resources, Authentication, Authorization and Password Policy and Audit Trail.

In Part 2, we will continue reviewing additional best practices for building new environments in the cloud.

## Budget Control

It is crucial to set a budget and budget alerts for any account in the cloud at in the early stages of working with in cloud environment. This is important in order to avoid scenarios in which high resource consumption happens due to human error, such as purchasing or consuming expensive resources, or of Denial of Wallet scenarios, where external attackers breach an organization's cloud account and deploys servers for Bitcoin mining.

Common examples of budget control management for various cloud providers:

- AWS Consolidated Billing – Configure central account among all the AWS account in the organization, in-order to forward billing data (which will be accessible only for a limited amount of user accounts).
- GCP Cloud Billing Account – Central repository for storing all billing data from all GCP projects.

- Azure Cost Management – An interface for configuring budget and budget alerts for all Azure subscriptions in the organization. It is possible to consolidate multiple Azure subscriptions to Management Groups in-order to centrally control budgets for all subscriptions.
- Budget on Oracle Cloud Infrastructure – An interface for configuring budget and budget alerts for all compartments.

## Secure access to cloud environments

In order to avoid inbound access from the Internet to resources in cloud environments (virtual servers, databases, storage, etc.), it is highly recommended to deploy a bastion host, which will be accessible from the Internet (SSH or RDP traffic) and will allow access and management of resources inside the cloud environment.

Common guidelines for deploying Bastion Host:

- Linux Bastion Hosts on AWS
- Create an Azure Bastion host using the portal
- Securely connecting to VM instances on GCP
- Setting Up the Basic Infrastructure for a Cloud Environment, based on Oracle Cloud

The more we expand the usage of cloud environments, we can consider deploying a VPN tunnel from the corporate network (Site-to-site VPN) or allow client VPN access from the Internet to the cloud environment (such as AWS Client VPN endpoint, Azure Point-to-Site VPN, Oracle Cloud SSL VPN).

## Managing compute resources (Virtual Machines and Containers)

When selecting to deploy virtual machines in cloud environment, it is highly recommended to follow the following guidelines:

- Choose an existing image from a pre-defined list in the cloud providers' marketplace (operating system flavor, operating system build, and sometimes an image that includes additional software inside the base image).
- Configure the image according to organizational or application demands.
- Update all software versions inside the image.
- Store an up-to-date version of the image ("Golden Image") inside the central image repository in the cloud environment (for reuse).
- In case the information inside the virtual machines is critical, consider using managed backup services (such as AWS Backup or Azure Backup).
- When deploying Windows servers, it is crucial to set complex passwords for the local Administrator's account, and when possible, join the Windows machine to the corporate domain.
- When deploying Linux servers, it is crucial to use SSH Key authentication and store the private key(s) in a secure location.

- Whenever possible, encrypt data at rest for all block volumes (the server's hard drives / volumes).
- It is highly recommended to connect the servers to a managed vulnerability assessment service, in order to detect software vulnerabilities (services such as Amazon Inspector or Azure Security Center).
- It is highly recommended to connect the servers to a managed patch management service in-order to ease the work of patch management (services such as AWS Systems Manager Patch Manager, Azure Automation Update Management or Google OS Patch Management).

When selecting to deploy containers in the cloud environment, it is highly recommended to follow the following guidelines:

- Use a Container image from a well know container repository.
- Update all binaries and all dependencies inside the Container image.
- Store all Container images inside a managed container repository inside the cloud environment (services such as Amazon ECR, Azure Container Registry, GCP Container Registry, Oracle Cloud Container Registry, etc.)
- Avoid using Root account inside the Containers.
- Avoid storing data (such as session IDs) inside the Container – make sure the container is stateless.
- It is highly recommended to connect the CI/CD process and the container update process to a managed vulnerability assessment service, in-order to detect software vulnerabilities (services such as Amazon ECR Image scanning, Azure Container Registry, GCP Container Analysis, etc.)

## Storing sensitive information

It is highly recommended to avoid storing sensitive information, such as credentials, encryption keys, secrets, API keys, etc., in clear text inside virtual machines, containers, text files or on the local desktop.

Sensitive information should be stored inside managed vault services such as:

- AWS KMS or AWS Secrets Manager
- Azure Key Vault
- Google Cloud KMS or Google Secret Manager
- Oracle Cloud Infrastructure Key Management
- HashiCorp Vault

**About the author**

Eyal Estrin is a cloud architect, working in the Inter-University Computation Center (IUCC) in Israel. He has more than 20 years of experience in infrastructure, information security and public cloud services. He is a public columnist and shares knowledge about cloud services. You can follow him on Twitter at @eyalestrin

# Getting Started: Best Practices for Deploying New Environments in the Cloud for the First Time – Part 3




July 16, 2020 Best Practices for Building First-Time Cloud Deployment Audrey Gerber

In Part 1 and Part 2 of this three-part blog series, we reviewed Resource Allocation Planning, Tagging Resources, Authentication, Authorization and Password Policy, Audit Trail, Budget Control, Secure Access to cloud environments, Managing Compute Resources and Storing Sensitive Information.

In the third, and final part of the series, we review additional best practices for building new environments in the cloud.

## Object Storage

When using Object Storage, it is recommended to follow the following guidelines:

- Avoid allowing public access to services such as Amazon S3, Azure Blob Storage, Google Cloud Storage, Oracle Cloud Object Storage, etc.
- Enable audit access on Object Storage and store the access logs in a central account in the cloud environment (which will be accessible only for a limited amount of user accounts).
- It is highly recommended to encrypt data at rest on all data inside Object Storage and when there is a business or regulatory requirement, and encrypt data using customer managed keys.
- It is highly recommended to enforce HTTPS/TLS for access to object storage (users, computers and applications).

- Avoid creating object storage bucket names with sensitive information, since object storage bucket names are unique and saved inside the DNS servers worldwide.

## Networking

- Make sure access to all resources is protected by access lists (such as AWS Security Groups, Azure Network Security Groups, GCP Firewall Rules, Oracle Cloud Network Security Groups, etc.)
- Avoid allowing inbound access to cloud environments using protocols such as SSH or RDP (in case remote access is needed, use Bastion host or VPN connections).
- As much as possible, it is recommended to avoid outbound traffic from the cloud environment to the Internet. If needed, use a NAT Gateway (such as Amazon NAT Gateway, Azure NAT Gateway, GCP Cloud NAT, Oracle Cloud NAT Gateway, etc.)
- As much as possible, use DNS names to access resources instead of static IPs.
- When developing cloud environments, and subnets inside new environments, avoid IP overlapping between subnets in order to allow peering between cloud environments.

## Advanced use of cloud environments

- Prefer to use managed services instead of manually managing virtual machines (services such as Amazon RDS, Azure SQL Database, Google Cloud SQL, etc.).
  It allows consumption of services, rather than maintaining servers, operating systems, updates/patches, backup and availability, assuming managed services in cluster or replica mode is chosen.

- Use Infrastructure as a Code (IoC) in-order to ease environment deployments, lower human errors and standardize deployment on multiple environments (Prod, Dev, Test).
  Common Infrastructure as a Code alternatives:

    - HashiCorp Terraform
    - AWS CloudFormation
    - Azure Resource Manager
    - Google Cloud Deployment Manager
    - Oracle Cloud Resource Manager

## Summary

To sum up:

Plan. Know what you need. Think scale.

If you use the best practices outlined here, taking off to the cloud for the first time will be an easier, safer and smoother ride then you might expect.

# Additional references

- AWS Well-Architected
- Microsoft Azure Well-Architected Framework
- Google Cloud's Architecture Framework
- Oracle Cloud Infrastructure Best Practices Framework

**About the author**

Eyal Estrin is a cloud architect, working in the Inter-University Computation Center (IUCC) in Israel. He has more than 20 years of experience in infrastructure, information security and public cloud services. He is a public columnist and shares knowledge about cloud services. You can follow him on Twitter at @eyalestrin