Tony Shen

AWS FOUNDATION BUILD NAMING CONVENTION

Contents

Revision: v0.2
November 16, 2019

Glossary

Term	Description
AWS	Amazon Web Services
Agile	Rapid development strategy by adaptive planning, quick response to changes, evolutionary software development approaches and collaboration across functional teams, customers and users
CI/CD	Continuous Integration / Continuous Delivery or Deployment
VPC	Virtual Private Cloud
CIDR	Classless Inter-Domain Routing
AZ	Availability Zone
FB	Foundation Build
NC	Naming Convention
CR	Created Resource

Introduction

Naming convention (NC) is an elaborated way by design in formulating how to name virtual constructs in a cloud like AWS. This article presents a NC with a focus on Foundation Build (FB) in AWS. FB refers to a collection of basic operational constructs created, deployed and running in AWS, forming a fundamental virtual infrastructure upon which AWS services can be effectively used.

In order to manage, operate, and develop such an infrastructure efficiently in AWS, a sensible and solid NC is required. When using NC, standards need to be followed to name numerous constructs derived from various AWS services in the infrastructure. By NC, constructs are named so that they are easily identifiable with respect to what they are, where they are, what they do, and what relationship they have with one another. NC needs to be extensible, meaning no matter how many more constructs will be added to the infrastructure over time in future it remains consistent, applicable, and easy to use. NC facilitates cross-referencing constructs, thereby helping to a great extent managing, using and developing your AWS cloud environment, particularly when you are doing it the Agile and CI/CD way by code.

This article introduces a NC as a candidate for you to consider adopting. You may also use it as a baseline to build up a better NC of your own that fits all of your needs.

Concepts, Definitions, and Terms

The NC is based on the following concepts, definitions, and associated terms

- NC is applicable to Created Resources (CRs) only
- A Created Resource (CR) is a construct instantiated from an AWS service
- AWS services are AWS native services available to use in AWS cloud, not including services based on any product(s) from AWS Marketplace or from third parties

AWS Services

As of this writing, AWS offers 143 services. Those services can be found in 23 categories.

1. Compute
2. Storage
3. Database
4. Migration & Transfer
5. Networking & Content Delivery
6. Developer Tools
7. Analytics
8. Robotics

9. Blockchain
10. Satellite
11. Management & Governance
12. Media Services
13. Machine Learning
14. Security, Identity & Compliance
15. Mobile
16. AR & VR
17. Application Integration
18. AWS Cost Management
19. Customer Engagement
20. Business Applications
21. Desktop & App Streaming
22. Internet of Things
23. Game Development

Most Commonly Used AWS Services

Of those categories listed above, nine (9) categories are the most commonly used ones.

1. Compute
2. Storage
3. Database
4. Networking & Content Delivery
5. Developer Tools
6. Management & Governance
7. Security, Identity & Compliance
8. Application Integration
9. AWS Cost Management

Those nine service categories can be further divided into two groups, one is of Phase 1 group; the other, Phase 2.

When using cloud, companies and organizations don't use all the services at once. Instead they start with most basic ones in certain categories and gradually move on to expand their scope by bringing more services into use when needed. In other words, services are getting used in phases. In the initial phase or Phase 1, only the most basic services are used, followed by more sophisticated or specialized services in subsequent phases.

Table 1 below shows the nine service categories in two phases. Phase 1 is the initial phase. Phase 2 is any time frame that you may advance into after the initial phase. It is the typical path that most companies and organizations take.

Table 1 - Phases

Service Category	Phase	Comment
Networking & Content Delivery	Phase 1	Planning and allocating virtual resources
Compute	Phase 1	Setting up instances (virtual computers)
Storage	Phase 1	Planning, allocating, using, managing virtual storage resources
Database	Phase 1	Planning, using, managing database services
Security, Identity & Compliance	Phase 1	Setting up users, groups, and roles
Management & Governance	Phase 1	Managing services
Developer Tools	Phase 2	Developing Cloud based apps
Application Integration	Phase 2	Integrating apps
Analytics	Phase 2	Analyzing data for auditing, security, and improvement

In each category, certain services are used more often than others. Table 2 below lists such services and their respective categories.

Table 2 - Most Commonly Used Services and Categories

Service	Service Category	Phase
EC2 VPC	Networking & Content Delivery	Phase 1
Route 53	Networking & Content Delivery	Phase 1
CloudFront	Networking & Content Delivery	Phase 1
EC2	Compute	Phase 1
Lambda	Compute	Phase 1
EKS	Compute	Phase 1
ECS	Compute	Phase 1
ECR	Compute	Phase 1
S3	Storage	Phase 1
S3 Glacier	Storage	Phase 1
RDS	Database	Phase 1
AWS Redshift	Database	Phase 1
DynamoDB	Database	Phase 1
CloudFormation	Management & Governance	Phase 1
AWS Organizations	Management & Governance	Phase 1
Control Tower	Management & Governance	Phase 1
CloudWatch	Management & Governance	Phase 1
CloudTrail	Management & Governance	Phase 1
Systems Manager	Management & Governance	Phase 1
Service Catalog	Management & Governance	Phase 1
Config	Management & Governance	Phase 1
AWS Auto Scaling	Management & Governance	Phase 1

IAM	Security, Identity & Compliance	Phase 1
Secrets Manager	Security, Identity & Compliance	Phase 1
Certificate Manager	Security, Identity & Compliance	Phase 1
Resource Access Manager	Security, Identity & Compliance	Phase 1
AWS Single Sign-On	Security, Identity & Compliance	Phase 1
Key Management Service	Security, Identity & Compliance	Phase 1
Simple Notification Service (SNS)	Application Integration	Phase 2
CodeStar	Developer Tools	Phase 2
CodeCommit	Developer Tools	Phase 2
CodeBuild	Developer Tools	Phase 2
CodeDeploy	Developer Tools	Phase 2
CodePipeline	Developer Tools	Phase 2
Athena	Analytics	Phase 2

FB typically involves the services listed above, but not necessarily all of them.

AWS Service Resource Types

Each AWS Service has its own Resource Type(s) for creating resources that it governs. Created resources (RCs) are determined by Resource Types for what they are, what they do, and how they work in AWS. Table 3 below shows a number of Resource Types under EC2 Service

Table 3 - EC2 Resource Types that Support Tags

Resource Type	Description
AWS::EC2::CustomerGateway	Customer Gateway used for VPN
AWS::EC2::DHCPOptions	DHCP Options
AWS::EC2::Instance	EC2 Instance
AWS::EC2::InternetGateway	Internet Gateway
AWS::EC2::NatGateway	NAT Gateway
AWS::EC2::NetworkAcl	Network ACL
AWS::EC2::NetworkInterface	Network Interface for Instances and Subnets
AWS::EC2::RouteTable	Route Table
AWS::EC2::SecurityGroup	Security Group
AWS::EC2::Subnet	Subnet
AWS::EC2::TrafficMirrorFilter	Traffic Mirror Filter for multi-account network monitoring
AWS::EC2::TrafficMirrorSession	Traffic Mirror Session multi-account network monitoring
AWS::EC2::TrafficMirrorTarget	Traffic Mirror Target multi-account network monitoring
AWS::EC2::TransitGateway	Transit Gateway for inter-connecting VPCs
AWS::EC2::TransitGatewayAttachment	Transit Gateway Attachment
AWS::EC2::TransitGatewayRouteTable	Transit Gateway Route Table
AWS::EC2::Volume	EBS storage volume attached to EC2 instance
AWS::EC2::VPC	Virtual Private Cloud

AWS::EC2::VPCPeeringConnection	VPC Peering Connection
AWS::EC2::VPNConnection	VPN Connection
AWS::EC2::VPNGateway	VPN Gateway

AWS Regions and Availability Zones

In addition to AWS services and Resource Types, another thing that bears heavily on NC is AWS regions and Availability Zones.

What is AWS region? An AWS region is part of AWS global network infrastructure. AWS infrastructure covers dozens of geographical service regions. As of this writing, AWS has 25 regions worldwide. Table 4 below lists those regions. Be aware that not all AWS services are available in every region. In other words, some regions have more services available than other regions.

Table 4 – AWS Regions

Region Name	Region
US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Hong Kong)	ap-east-1
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka-Local)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2

Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
China (Beijing)	cn-north-1
China (Ningxia)	cn-northwest-1
EU (Frankfurt)	eu-central-1
EU (Ireland)	eu-west-1
EU (London)	eu-west-2
EU (Paris)	eu-west-3
EU (Stockholm)	eu-north-1
Middle East (Bahrain)	me-south-1
South America (Sao Paulo)	sa-east-1
AWS GovCloud (US-East)	us-gov-east-1
AWS GovCloud (US-West)	us-gov-west-1

In each region, AWS maintains multiple data centers. Each data center serves as an Availability Zone (AZ) in AWS. See Table 5 below for a list of AZs in four US regions

Table 5 – AZs in US Regions

Region Name	Region Code	AZ Code
US East (N. Virginia)	us-east-1	us-east-1a
US East (N. Virginia)	us-east-1	us-east-1b
US East (N. Virginia)	us-east-1	us-east-1c
US East (N. Virginia)	us-east-1	us-east-1d

US East (N. Virginia)	us-east-1	us-east-1e
US East (N. Virginia)	us-east-1	us-east-1f
US East (Ohio)	us-east-2	us-east-2a
US East (Ohio)	us-east-2	us-east-2b
US East (Ohio)	us-east-2	us-east-2c
US West (N. California)	us-west-1	us-west-1a
US West (N. California)	us-west-1	us-west-1b
US West (Oregon)	us-west-2	us-west-2a
US West (Oregon)	us-west-2	us-west-2b
US West (Oregon)	us-west-2	us-west-2c
US West (Oregon)	us-west-2	us-west-2d

The number of AZs is not identical across the regions. In Table 5 above, for instance, N. Virginia region boasts of as many as six AZs whereas N. California region has only two AZs.

AZs are designed to provide high availability (HA). With AWS AZs in place it is possible for your applications to run continuously without interruption in event of failure that occurs in any single AZ in the region.

AWS Virtual Boundaries

When you visit a data center, you won't have difficulty in seeing its physical boundary with walls, buildings, security entrances and exits. When you go online using a public cloud like AWS, you don't see such physical barriers. Instead AWS has virtual boundaries in place to keep your cloud space private and secure. AWS virtual boundaries come in three forms, AWS account, AWS Organization, and CIDR.

AWS Account

Your AWS account defines your first boundary. You sign in with your AWS account (See Getting Started with AWS for more details), and in that account of yours what you see and do does not

interfere with any other customers and vice versa. What you have in your account is not visible to other customers either, and vice versa. This is called tenancy isolation, like what suites do for tenants in an office building.

AWS Organization

A second boundary is established when you have multiple AWS accounts for your subsidiaries and/or departments to use. AWS let you to organize your accounts in a virtual entity called "Organization". In your organization, you can do more with your accounts, not only with each account individually but also with all accounts collectively, like what you can get a certain amount of synergy out of your combined resources when you rented a number of suites or floors in the office building. AWS maintains tenancy isolation and separation for your organization, too. Your Organization is isolated from other customers, no matter whether other customers have their organizations.

CIDR

Within an AWS account, a third boundary exists for you to use to isolate and separate your networks internally. This boundary is based on Classless Inter-Domain Routing (CIDR). CIDR primarily applies to VPC (Virtual Private Cloud). As a network construct, VPC contains subnets. As part of VPC, subnets are defined in CIDR, too. CIDR is essentially an IP address range. VPCs live in their respective IP ranges, isolated from each other. You can view CIDR(s) as interior walls or panels for you to use to partition your virtual network space in your AWS account(s).

AWS Service Scope

In designing a NC, AWS service scope plays an important role. In AWS, services are scoped in two groups. One is regional services; the other, global services.

Regional Services

VPC and a good number of other AWS services are regional services, meaning those services are scoped within a region and region-dependent. Specifically, if a resource was created from a regional service in a region, that resource exists only in that region. For example, if you create a VPC in N. Virginia region, that VPC of yours stays and runs only in N. Virginia. Table 6 below lists commonly used regional services

Table 6 - Most Commonly Used Regional Services

Service	Service Category	Phase

EC2 VPC	Networking & Content Delivery	Phase 1
EC2	Compute	Phase 1
Lambda	Compute	Phase 1
EKS	Compute	Phase 1
ECS	Compute	Phase 1
ECR	Compute	Phase 1
RDS	Database	Phase 1
AWS Redshift	Database	Phase 1
DynamoDB	Database	Phase 1
CloudFormation	Management & Governance	Phase 1
Control Tower	Management & Governance	Phase 1
CloudWatch	Management & Governance	Phase 1
CloudTrail	Management & Governance	Phase 1
Systems Manager	Management & Governance	Phase 1
Config	Management & Governance	Phase 1
AWS Auto Scaling	Management & Governance	Phase 1
Secrets Manager	Security, Identity & Compliance	Phase 1
Certificate Manager	Security, Identity & Compliance	Phase 1
Resource Access Manager	Security, Identity & Compliance	Phase 1
CodeStar	Developer Tools	Phase 2
CodeCommit	Developer Tools	Phase 2
CodeBuild	Developer Tools	Phase 2
CodeDeploy	Developer Tools	Phase 2
CodePipeline	Developer Tools	Phase 2
Athena	Analytics	Phase 2

Global Services

Other than regional services, there are Global services. A global service is not confined in one region. A resource created from a global service is available and applies to all regions. One such an example is IAM user. When you create a user in your account, that user exists in all regions in your account. Table 7 lists a number of commonly used global services.

Table 7 - Most Commonly Used Global Services

Service	Service Category	Phase
Route 53	Networking & Content Delivery	Phase 1
S3	Storage	Phase 1
S3 Glacier	Storage	Phase 1
AWS Organizations	Management & Governance	Phase 1
Service Catalog	Management & Governance	Phase 1
IAM	Security, Identity & Compliance	Phase 1
AWS Single Sign-On	Security, Identity & Compliance	Phase 1

Key Management Service	Security, Identity & Compliance	Phase 1
Simple Notification Service (SNS)	Application Integration	Phase 2

Naming Convention Foundation Cornerstones

So far we have touched AWS Services, Regions and Availability Zones, AWS Virtual Boundaries, and AWS Service Scope. These are four important aspects of AWS. For NC to work well, cornerstones need to be planted in to solidify those dimensions, so to speak. The cornerstones are a number of code tables. The code tables code AWS services and other entities so that they can be referred to by coded identifiers instead of their full names. The code tables were intended to shorten a construct's name that needs to include those identifiers. The code tables are presented in Table 8, 9, 10, 11, 12, and 13 below.

NC Code Tables

Table 8 – AWS Service Code Table

Service	Service Category	R/G	NC Code
EC2	Compute	R	rec2
Lambda	Compute	R	rlam
EKS	Compute	R	reks
ECS	Compute	R	recs
ECR	Compute	R	recr
RDS	Database	R	reds
AWS Redshift	Database	R	rred
DynamoDB	Database	R	rdyn
CloudFormation	Management & Governance	R	rcfn
Control Tower	Management & Governance	R	rtwr
CloudWatch	Management & Governance	R	rclw
CloudTrail	Management & Governance	R	rclt
Systems Manager	Management & Governance	R	rssm
Config	Management & Governance	R	rcfg
AWS Auto Scaling	Management & Governance	R	rasc
Secrets Manager	Security, Identity & Compliance	R	rscm
Certificate Manager	Security, Identity & Compliance	R	rctm
Resource Access Manager	Security, Identity & Compliance	R	rram
CodeStar	Developer Tools	R	rcst
CodeCommit	Developer Tools	R	rccm
CodeBuild	Developer Tools	R	rcbd
CodeDeploy	Developer Tools	R	rcdp
CodePipeline	Developer Tools	R	rcpl

Athena	Analytics	R	rath
Route 53	Networking & Content Delivery	G	gr53
S3	Storage	G	gs3b
S3 Glacier	Storage	G	gs3g
AWS Organizations	Management & Governance	G	gorg
Service Catalog	Management & Governance	G	gscl
IAM	Security, Identity & Compliance	G	giam
AWS Single Sign-On	Security, Identity & Compliance	G	gsso
Key Management Service	Security, Identity & Compliance	G	gkms
Simple Notification Service (SNS)	Application Integration	G	gsns

Table 9 – AWS Service Resource Types Code Table (EC2)

Resource Type	Description	NC Code
AWS::EC2::CustomerGateway	Customer Gateway used for VPN	ec2cgw
AWS::EC2::DHCPOptions	DHCP Options	ec2dhc
AWS::EC2::Instance	EC2 Instance	ec2ins
AWS::EC2::InternetGateway	Internet Gateway	ec2igw
AWS::EC2::NatGateway	NAT Gateway	ec2ngw
AWS::EC2::NetworkAcl	Network ACL	ec2acl
AWS::EC2::NetworkInterface	Network Interface for Instances and Subnets	ec2int
AWS::EC2::RouteTable	Route Table	ec2rtb
AWS::EC2::SecurityGroup	Security Group	ec2sgp
AWS::EC2::Subnet	Subnet	ec2sub
AWS::EC2::TrafficMirrorFilter	Traffic Mirror Filter for multi-account network monitoring	ec2tmf
AWS::EC2::TrafficMirrorSession	Traffic Mirror Session multi-account network monitoring	ec2tms
AWS::EC2::TrafficMirrorTarget	Traffic Mirror Target multi-account network monitoring	ec2tmt
AWS::EC2::TransitGateway	Transit Gateway for inter-connecting VPCs	ec2tgw
AWS::EC2::TransitGatewayAttachment	Transit Gateway Attachment	ec2tga
AWS::EC2::TransitGatewayRouteTable	Transit Gateway Route Table	ec2tgr
AWS::EC2::Volume	EBS storage volume attached to EC2 instance	ec2vol
AWS::EC2::VPC	Virtual Private Cloud	ec2vpc
AWS::EC2::VPCPeeringConnection	VPC Peering Connection	ec2vpe
AWS::EC2::VPNConnection	VPN Connection	ec2vpn
AWS::EC2::VPNGateway	VPN Gateway	ec2vgw

Table 10 – AWS Regions and Availability Zones Code Table

Region Name	AWS Region Code	NC Code	AWS AZ Code	NC Code
US East (N. Virginia)	us-east-1	use1	us-east-1a	use1-a
US East (N. Virginia)	us-east-1	use1	us-east-1b	use1-b
US East (N. Virginia)	us-east-1	use1	us-east-1c	use1-c
US East (N. Virginia)	us-east-1	use1	us-east-1d	use1-d
US East (N. Virginia)	us-east-1	use1	us-east-1e	use1-e
US East (N. Virginia)	us-east-1	use1	us-east-1f	use1-f
US East (Ohio)	us-east-2	use2	us-east-2a	use2-a
US East (Ohio)	us-east-2	use2	us-east-2b	use2-b
US East (Ohio)	us-east-2	use2	us-east-2c	use2-c
US West (N. California)	us-west-1	usw1	us-west-1a	usw1-a
US West (N. California)	us-west-1	usw1	us-west-1b	usw1-b
US West (Oregon)	us-west-2	usw2	us-west-2a	usw2-a
US West (Oregon)	us-west-2	usw2	us-west-2b	usw2-b
US West (Oregon)	us-west-2	usw2	us-west-2c	usw2-c
US West (Oregon)	us-west-2	usw2	us-west-2d	usw2-d
All Regions *	N/A	allr	N/A	allr-a

All Regions code applies to Global Services

Table 11 – Virtual Boundary Organization Code Table

Organization	Description	NC Code
Wholesale	Wholesale management, operations, support	O1
Retail	Retail management, operations, support	O2
Lending	Commercial and consumer Lending management, operations, support	O3
Investment	Corporate and individual investment management, operations, support	O4
Credit Card	Credit card management, operations, support	O5

Table 12 – Virtual Boundary Account Code Table

Organization	Account	Account Owner	NC Code
Wholesale	123456789012	Marketing	o1-a01
Retail	123456789022	Customer Service	o2-a02
Lending	123456789032	Legal	o3-a03
Investment	123456789042	Financing	o4-a04
Credit Card	123456789052	Administration	o5-a05

Table 13 – Environment Code Table

Environment Name	Description	NC Code
Dev	Development	dev
Prod	Production	prd
Test	User Acceptance Testing	uat
Staging	Staging	stg
All	All environments	all

With the code tables laid out, let's move on and see how a name tag is composed of, and what it looks like. A name tag has its prefix, ID, and postfix.

NC Name Tag Components

Name Tag Prefix

For Regional Service Created Resources (CRs), the prefix is made up with the following:
Organization Code + Account Code + Region Code + Service Code + Resource Type Code + Environment Code.

Org Code	Acct Code	Region Code	Service Code	Resource Type Code	Environment Code

Example: **o1-a01-use1-rec2ins-dev** (EC2 Instance in Organization 1, Account 1, US N. Virginia region, in Development)

o1	a01	use1	rec2	ins	dev

For Global Service Created Resources (CRs), the prefix is made up with the following:
Organization Code + Account Code + All Regions Code + Service Code + Resource Type Code + Environment Code

Org Code	Acct Code	All Regions Code	Service Code	Resource Type Code	Environment Code

Example: **o1-a01-allr-giamusr-prd** (IAM user in Organization 1, Account 1, All Regions, in Production)

o1	a01	allr	giam	usr	prd

Name Tag ID

Note: ID is an arbitrary number used to differentiate one RC from another. ID needs to be in fixed length in order to facilitate sorting, filtering and cross-referencing by code

Name Tag Postfix

Name tag postfix is a varied length string that provides additional mnemonic description of an CR.

Example:
mkt-vpc1-2, meaning that the RC was for Marketing's VPC1 and VPC2.

Name Tag Format

Prefix + ID + Postfix

Example 1 - Regional Service CR - **o1-a01-use1-rec2cgw-dev-01-mkt-vpc1-2**

Prefix (org-acct-region-resourcetype-env-id	**ID**	**Postfix (varied length String)**
o1-a01-use1-rec2cgw-dev-	**001-**	mkt-vpc1-2

In Example 1 above, the name tag indicates that the RC is a custom gateway attached to VPC1 and VPC2 used by Marketing in Development in AWS Account 01 in Organization 01.

Example 2 – Global Service CR - **o1-a01-allr-giamgrp-dev-001-admins**

Prefix (org-acct-allr-resourcetype-env-id	**ID**	**Postfix (varied length String)**
o1-a01-allr-giamgrp-dev-001	**001-**	admins

In Example 2 above, the name indicates that CR is a Cloud admins user group in Development in Account 01 in Organization 01

When going online interacting with AWS, either by Console, AWS CLI, or AWS SDK, Organization, Account, and Region are known whereas when being offline such information is not present.

Taking into account of this difference this NC includes Organization, Account, and Region indicators in every RC name tag by design thereby helping with use cases such as offline auditing on downloaded logs from AWS and integration with on-premise based management tools that receive log feeds from AWS, just name a few. Another strong use case for this design is for centralized management of multiple Organizations.

NC in Action

As it was stated at the beginning of this article, the NC in discussion was to apply to RCs. Since a RC is created from a Resource Type in an AWS Service, in this section, we inspect a few figurative RC names that follow the NC standards. Those RCs are listed by AWS Service and Resource Type. There are a number of Regional Service RCs, and a few of Global Service RCs.

You may use the RC names as examples to apply to other services and resource types. NC is supposed to apply to all Services and Resource Types with only one caveat - a Resource Type has to either have a Name type property or support tags. Be aware that not all Resource Types do.

Regional Service CRs

Regional Service Athena

Table 10 – Athena Resource Types that Support Name

Resource Type	NC Code	Name Example
AWS::Athena::NamedQuery	athqry	**o1-a01-allr-rathqry-dev-001**

Regional Service Auto Scaling

No resource type that supports tags in this service

Regional Service Certificate Manager

Table 11 – Certificate Manager Resource Types that Support Tags

Resource Type	NC Code	Name Tag Example
AWS::CertificateManager::Certificate	cmgcrt	**o1-a01-use1-rcmgcrt-dev-001**

Regional Service CloudFormation

Table 12 – CloudFormation Resource Types that Support Tags

Resource Type	NC Code	Name Tag Example
AWS::CloudFormation::Stack	cfnstk	**o1-a01-use1-rcfnstk-dev-001-lampForMarketing**

Regional Service EC2

Table 13 - EC2 Resource Types that Support Tags

Resource Type	NC Code	Name Tag Example
AWS::EC2::CustomerGateway	ec2cgw	**o1-a01-use1-rec2cgw-dev-001**
AWS::EC2::DHCPOptions	ec2dhc	**o1-a01-use1-rec2dhc-dev-001**
AWS::EC2::Instance	ec2ins	**o1-a01-use1-rec2ins-dev-001**
AWS::EC2::InternetGateway	ec2igw	**o1-a01-use1-rec2igw-dev-001**
AWS::EC2::NatGateway	ec2ngw	**o1-a01-use1-rec2ngw-dev-001**
AWS::EC2::NetworkAcl	ec2acl	**o1-a01-use1-rec2acl-dev-001**
AWS::EC2::NetworkInterface	ec2int	**o1-a01-use1-rec2int-dev-001**
AWS::EC2::RouteTable	ec2rtb	**o1-a01-use1-rec2rtb-dev-001**
AWS::EC2::SecurityGroup	ec2sgp	**o1-a01-use1-rec2sgw-dev-001**
AWS::EC2::Subnet	ec2sub	**o1-a01-use1-rec2sub-dev-001 ***
AWS::EC2::TrafficMirrorFilter	ec2tmf	**o1-a01-use1-rec2tmf-dev-001**
AWS::EC2::TrafficMirrorSession	ec2tms	**o1-a01-use1-rec2tms-dev-001**
AWS::EC2::TrafficMirrorTarget	ec2tmt	**o1-a01-use1-rec2tmt-dev-001**
AWS::EC2::TransitGateway	ec2tgw	**o1-a01-use1-rec2tgw-dev-001**
AWS::EC2::TransitGatewayAttachment	ec2tga	**o1-a01-use1-rec2tgw-dev-001**
AWS::EC2::TransitGatewayRouteTable	ec2tgr	**o1-a01-use1-rec2tgr-dev-001**
AWS::EC2::Volume	ec2vol	**o1-a01-use1-rec2vol-dev-001**
AWS::EC2::VPC	ec2vpc	**o1-a01-use1-rec2vpc-dev-001**
AWS::EC2::VPCPeeringConnection	ec2vpe	**o1-a01-use1-rec2vpe-dev-001**
AWS::EC2::VPNConnection	ec2vpn	**o1-a01-use1-rec2vpn-dev-001**
AWS::EC2::VPNGateway	ec2vgw	**o1-a01-use1-rec2vgw-dev-001**

Note: When needed, Availability Zone (AZ) where Subnet is running can be indicated in postfix, though it is not advised to do so because Subnet has AZ property by which its AZ association can be easily found by Console or by code. Furthermore, Subnet does not change its AZ after its creation.

Regional Service EC2 Auto Scaling

Table 14 – EC2 Auto Scaling Resource Types that Support Tags

Resource Type	NC Code	Name Tag Example
AWS::AutoScaling::AutoScalingGroup	atsgrp	**o1-a01-use1-ratsgrp-dev-001**

Regional Service CloudTrail

Table 15 – CloudTrail Resource Types that Support Tags

Resource Type	NC Code	Name Tag Example
AWS::CloudTrail::Trail	clttrl	**o1-a01-use1-rctltrl-dev-001**

Global Service CRs

Global Service IAM

Table 16 – IAM Resource Types that Support Name

Resource Type	NC Code	Name Example
AWS::IAM::Group (GroupName)	iamgrp	**o1-a01-allr-giamgrp-dev-001**
AWS::IAM::Role (RoleName)	iamrol	**o1-a01-allr-giamrol-dev-001**
AWS::IAM::User (UserName)	iamusr	**o1-a01-allr-giamusr-dev-001**
AWS::IAM::Policy (PolicyName)	iampol	**o1-a01-allr-giampol-dev-001**

Table 17 – IAM Resource Types that Support Tags

Resource Type	NC Code	Name Tag Example
AWS::IAM::Role	iamrol	**o1-a01-allr-giamrol-dev-001**

CR Set Examples

CR Set refers to a set of CRs created to deliver a particular function. Assign a name to the set. Each CR in the set bears the set name in its postfix. The postfix becomes a badge for all the CRs in the set, indicating a common purpose.

Example 1 – IAM Change Alert
The CR set detects IAM changes that includes user, group, role, policy creation, removal and updates and sends email alerts to cloud admins.

Multiple CRs need to be created to work together as a set to deliver this function. The required CRs are:

- SNS topic
- IAM policy
- IAM role
- Lambda function
- Cloud Watch Event Rule

Table 18 below lists the CR names:

Table 18 – CR Set Names

CR	Name
CR Set	IAMChgAlert
SNS topic	o1-a03-use1-rsnstop-all-999-IAMChgAlert
IAM policy	o1-a03-allr-giampol-all-999-IAMChgAlert
IAM role	o1-a03-allr-giamrol-all-999-IAMChgAlert
Lambda function	o1-a03-use1-rec2lmb-all-999-IAMChgAlert
CloudWatch Event Rule	o1-a03-use1-rclwrul-all-999-IAMChgAlert

Simplified Naming Convention (SNC)

Where SNC Simplifies

NC can be simplified by coding the prefix.

The prefix includes four elements

1. Organization ID
2. Account ID
3. Region Code
4. Environment Code

These four elements can be folded into single coded element. Here is how

1. Using a single letter to code Organizations. The single letter designation allows up to 26 Organizations
2. Using a single letter to code Accounts. The single letter designation allows up to 26 Accounts
3. Using a single digit to code Regions. The single digit designation allows up to 10 regions
4. Using a single letter to code environments. The single letter designation allows up to 26 Environments

Table 19 below shows the coding for each element

Table 19 – Single-letter Organization Code table

Code	Organization	Code	Account	Code	Region	Code	Environment
A	Organization 1	A	Account 1	0	All regions	A	All environments
B	Organization 2	B	Account 2	1	US-East-1	D	Development
C	Organization 3	C	Account 3	2	US-East-2	P	Production
D	Organization 4	C	Account 4	3	US-West-1	Q	QA

E	Organization 5	C	Account 5	4	US-West-2	S	Staging

Table 20 below shows the new prefix coded by following SNC. The new prefix has only 4 characters whereas its old counterpart had as many 14 characters.

Table 20 - Organization 1 Account 3 All regions Development

Code	Organization	Code	Account	Code	Region	Code	Environment
a	Organization 1	c	Account 3	0	All regions	d	Development

ac0d (Organization 1 Account 3 All regions Development)

Example 1 – CR Set IAM Change Alert

Table 21 – CR Set Names that Follow SNC

CR	Name	
CR Set	IAMChgAlert	
SNS topic	ac1d-rsnstop-999-IAMChgAlert	Org1 Acct3 USE1 Development
IAM policy	ac0d-giampol-999-IAMChgAlert	Org1 Acct3 All regions Dev
IAM role	ac0d-giamrol-999-IAMChgAlert	Org1 Acct3 All regions Dev
Lambda function	ac1d-rec2lmb-999-IAMChgAlert	Org1 Acct3 USE1 Development
CloudWatch Event Rule	ac1d-rclwrul-999-IAMChgAlert	Org1 Acct3 USE1 Development

Update 1 (20191117)

Where Update 1 updated

Update 1 established Team Code

Table 22 - Team Code Table

Team/Department/Group	Code	
Networking	NT	
Compute	CP	
Storage	ST	
IAM	IA	
Security	SE	
Auditing	AU	
Developers	DV	

QA	QA	
Operations	OP	
Engineering	EG	
DBA	DB	
LOB Retail	RT	
LOB Wholesale	WH	
LOB Finance	FI	
LOB Marketing	MK	
LOB Investment	IN	
LOB Customer Service	CU	
LOB Commercial Lending	BL	
LOB Consumer Lending	CL	
LOB Credit Card	CR	

Team code is used to indicate the owner and user of services, and hence billing responsibility

Example 1 – IAM Change Alert CR Set

Table 23 – CR Set Names that Follow SNC Revised with Team Code

CR	**Name**	
CR Set	RTIAMChgAlert	Retail as owner and user
SNS topic	ac1d-rsnstop-RTIAMChgAlert	Org1 Acct3 USE1 Dev Retail
IAM policy	ac0d-giampol-RTIAMChgAlert	Org1 Acct3 All regions Dev Retail
IAM role	ac0d-giamrol-RTIAMChgAlert	Org1 Acct3 All regions Dev Retail
Lambda function	ac1d-rec2lmb-RTIAMChgAlert	Org1 Acct3 USE1 Dev Retail
CloudWatch Event Rule	ac1d-rclwrul-RTIAMChgAlert	Org1 Acct3 USE1 Dev Retail

Example 2 – Transit Gateway for Wholesale and Associated Services and CRs

Table 24 – CR Association Names that follow SNC Revised with Team Code

CR	Name	Description
CR Set	WHTGW1	Wholesale Transit Gateway 1
Transit Gateway	wa1d-rvpctgw-WHTGW1	Wholesale Transit Gateway 1
VPC	wa1d-rvpc086-WHVPCP01	Wholesale Prod VPC 1
VPC	wa1d-rvpc0de-WHVPCP02	Wholesale Prod VPC 2
Subnet	wa1d-rvpcsub-WHVPCP01SN01	Wholesale Prod VPC 1 Subnet 1
Subnet	wa1d-rvpcsub-WHVPCP02SN01	Wholesale Prod VPC 2 Subnet 1

TGW Attachment	wa1d-rvpctva-WHTGW1VPCP01SN01	Wholesale Transit Gateway Attachment from Prod VPC 1 Subnet 1
TGW Attachment	wa1d-rvpctva-WHTGW1VPCP02SN01	Wholesale Transit Gateway Attachment from Prod VPC 2 Subnet 1

Where “wa” stands for Organization Wholesale, Account 1

Summary

In this article we discussed one NC. In its design, we did the following:

- We limited our scope to two areas. We limited to primarily FB services or most commonly used services for simplicity. We limited to Name and Name Tag only, also for simplicity.
- We used code tables to abbreviate the names of AWS services and other entities. With the code tables NC uses abbreviated code names as identifiers. CR Name/Name tag includes those identifiers in its name’s prefix.
- In CR name/name tag, we use a numeric Name ID to differentiate one CR from another.
- CR Name/Name tag ends with a varied length string type of postfix for it to provide more mnemonic description of the CR.

This NC was intended to be applicable to CRs created from all AWS services and Resource Types as long as the resource type either has a Name property or supports tags.

This NC is also intended to support centralized management of Organizations and on-premise based management for AWS cloud.

References

AWS CloudFormation User Guide
AWS tagging strategies