

Cost Optimization on AWS Cloud - Part 1

Overview of AWS Services' Pricing and Cost Optimization

ဒီနေ့မှာတော့ကျွန်တော်တို့ Amazon Web Services အသုံးပြုနေကြတဲ့သူတွေအတွက် အသုံးဝင်လောက်မဲ့ articleလေး တစ်ခုကိုရေးပေးချင်ပါတယ်။ ကျွန်တော်တို့ ရုံးတွေ Organizationတွေ မှာ AWS သုံးကြတယ် Costတွေ ကလည်း တော်တော်လေးကုန် ကျနေတယ် ဘာတွေကကုန်နေမှန်းလည်း မသိဘူး Billing Console မှာတော့ပြနေတယ် ဘာတွေအတွက်ကဘယ်လောက် ဘယ်လောက်ကုန်နေတယ် ဒါပေမယ့် သူတို့ဘယ်လိုဘယ်လိုတွက်လိုက်လဲ မသိဘူးဖြစ်နေတယ်ဆိုရင် ၊ Cost တွေ ကအဆမတန်ကုန်ကျနေတယ်ဆိုရင် အခု article လေး က သင့်အတွက် လက်ရှိကုန်ကျနေတဲ့ Cost တွေရဲ့ တွက်ချက်ပုံ အကြမ်းဖျင်းနဲ့ Optimization သဘောတရား များကို အနည်းနဲ့အများ နားလည် သဘောပေါက်ပြီး အထောက်အကူပြုလိမ့်မယ်လို့ ယုံကြည်ပါတယ်။ ဟုတ်ပြီ အဲ့တော့ စလိုက်ကြရအောင်။

Foundation of Cost Optimization

ကျွန်တော်တို့ အနေနဲ့ ဘယ်လိုစလုပ်မလဲ

- အရင်ဆုံးကိုယ်က AWS ရဲ့ ဘယ် service တွေကို သုံးထားလဲသိရပါမယ် ဥပမာ - ဘယ် Region မှာ ဘာ service ကိုသုံးထားလဲပေါ့ ၊ North Virginia (us-east-1) မှာ EC2 သုံးထားတာလား Europe (Paris - eu-west-3)မှာ RDS သုံးထားတာလား စသဖြင့်ပေါ့
- ပြီးတော့ ကိုယ်သုံးထားတဲ့ service တွေရဲ့ pricing ကိုသေချာသိရပါမယ် ၊ ဒီမှာဘာပြသနာရှိလဲဆိုတော့ AWS က သူရဲ့ Region အလိုက် Service Pricing တွေက အများကြီးရှိတဲ့ အတွက်အကုန်တော့ ထည့်မပြောနိုင်တော့ပါဘူး။ ဥပမာအနေနဲ့ မြင်သာအောင်ပြောရမယ်ဆိုရင် North Virginia (us-east-1) မှာ ရှိတဲ့ EC2 Pricing နဲ့ Singapore (ap-southeast-1) မှာရှိတဲ့ EC2 Pricing က မတူပါဘူး။

Linux	RHEL	SLES	Windows	Windows with SQL Standard	Windows with SQL Web
Windows with SQL Enterprise	Linux with SQL Standard	Linux with SQL Web	Linux with SQL Enterprise		

Region:
US East (N. Virginia)

	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
a1.medium	1	N/A	2 GiB	EBS Only	\$0.0255 per Hour
a1.large	2	N/A	4 GiB	EBS Only	\$0.051 per Hour
a1.xlarge	4	N/A	8 GiB	EBS Only	\$0.102 per Hour
a1.2xlarge	8	N/A	16 GiB	EBS Only	\$0.204 per Hour
a1.4xlarge	16	N/A	32 GiB	EBS Only	\$0.408 per Hour
a1.metal	16	N/A	32 GiB	EBS Only	\$0.408 per Hour
t3.nano	2	Variable	0.5 GiB	EBS Only	\$0.0052 per Hour
t3.micro	2	Variable	1 GiB	EBS Only	\$0.0104 per Hour
t3.small	2	Variable	2 GiB	EBS Only	\$0.0208 per Hour
t3.medium	2	Variable	4 GiB	EBS Only	\$0.0416 per Hour
t3.large	2	Variable	8 GiB	EBS Only	\$0.0832 per Hour
t3.xlarge	4	Variable	16 GiB	EBS Only	\$0.1664 per Hour
t3.2xlarge	8	Variable	32 GiB	EBS Only	\$0.3328 per Hour
t3a.nano	2	Variable	0.5 GiB	EBS Only	\$0.0047 per Hour

US East (N. Virginia) EC2 t3.medium price per hour

Linux	RHEL	SLES	Windows	Windows with SQL Standard	Windows with SQL Web
Windows with SQL Enterprise	Linux with SQL Standard	Linux with SQL Web	Linux with SQL Enterprise		
Region: Asia Pacific (Singapore) ▾					
	vCPU	ECU	Memory (GiB)	Instance Storage (GB)	Linux/UNIX Usage
General Purpose - Current Generation					
a1.medium	1	N/A	2 GiB	EBS Only	\$0.0294 per Hour
a1.large	2	N/A	4 GiB	EBS Only	\$0.0588 per Hour
a1.xlarge	4	N/A	8 GiB	EBS Only	\$0.1176 per Hour
a1.2xlarge	8	N/A	16 GiB	EBS Only	\$0.2352 per Hour
a1.4xlarge	16	N/A	32 GiB	EBS Only	\$0.4704 per Hour
a1.metal	16	N/A	32 GiB	EBS Only	\$0.47 per Hour
t3.nano	2	Variable	0.5 GiB	EBS Only	\$0.0066 per Hour
t3.micro	2	Variable	1 GiB	EBS Only	\$0.0132 per Hour
t3.small	2	Variable	2 GiB	EBS Only	\$0.0264 per Hour
t3.medium	2	Variable	4 GiB	EBS Only	\$0.0528 per Hour
t3.large	2	Variable	8 GiB	EBS Only	\$0.1056 per Hour
t3.xlarge	4	Variable	16 GiB	EBS Only	\$0.2112 per Hour
t3.2xlarge	8	Variable	32 GiB	EBS Only	\$0.4224 per Hour
t3a.nano	2	Variable	0.5 GiB	EBS Only	\$0.0059 per Hour

Asia Pacific (Singapore) EC2 t3.medium price per hour

ဒီပုံမှာကြည့်မယ်ဆိုရင် US East (N. Virginia) EC2 t3.medium price per hour က **\$0.0416**
 ပဲကျသင့်မှာဖြစ်ပြီးတော့ Asia Pacific (Singapore) EC2 t3.medium price per hour က **\$0.0528**
 ကျသင့်မှာပါ။ တစ်ခြားသော Instance Type တွေလည်း ထိုနည်းလည်းကောင်းပါပဲ။ US East (N. Virginia) ကပိုသက်သာပါတယ်။ ဆိုလိုတာကတော့ ကျွန်တော်တို့တွေအနေနဲ့ ကိုယ်ရဲ့ company requirement အရ Region အလိုက်ဈေးနှုန်းတွေကို အကြမ်းဖျင်းနားလည်ထားရပါမယ်။ ဒါဆိုဘာလို့ US East (N. Virginia) ကပိုသက်သာတာလဲဆိုရင် AWS ရဲ့ Default Region ဖြစ်တာလည်း တစ်ကြောင်း၊ များပြားတဲ့ Customer တွေကို Service ပေးနိုင်ဖို့ Resource များများရှိတာရယ်၊ Physical Datacenter တွေအတွက် လိုအပ်တဲ့ Electrical Power ရပုံခြင်းမတူတာရယ် နိုင်ငံရဲ့ Tax rule ဘာညာအနေအထားတွေ မတူညီတာရယ်ကြောင့်လို့ ယေဘုယျ ဆိုနိုင်ပါတယ်။ Pricing တွေ ကို သဘောတရားကိုမြန်မြန်နားလည်နိုင်မှ အမြန်ဆုံးနည်းလမ်းကတော့ ကိုယ့် Manage လုပ်ရတဲ့ AWS

account ရဲ့ Billing Console ကို မနက်တိုင်းကြည့်ပြီး တစ်လလောက် Calculator တစ်လုံးဘေးချပြီးလိုက်တွက်ကြည့်ပါ။ (စာရေးသူ၏ ကိုယ်တွေ့မှတ်တမ်းက ကောက်နုတ်ချက်)

- အိုကေ ကိုယ်ကဘယ် Service တွေ ကိုသုံးထားတာလဲနဲ့ Pricing ကိုလည်း နားလည်နေပြီဆိုရင် နောက်တစ်ခု သိရမှာက ကိုယ်သုံးထားတဲ့ Service Resources တွေရဲ့ Workload ပါ။ ဆိုလိုတာက EC2 Workload တွေ CPU Utilization တွေ Memory Utilization တွေ Network Traffic တွေ ဘယ်အချိန်မှာဘယ်လောက်တော့ သုံးတယ် ဘယ်အချိန်မှာတော့ ဘယ်လောက်ပဲသုံးတော့တယ်ဆိုတာကို သိရပါမယ်။ အဲ့အတွက်လည်း ကိုယ်ရဲ့ Infrastructure သို့မဟုတ် ကိုယ့်ရဲ့ Resource ကို monitoring တွေ လုပ်ထားဖို့လိုပါတယ်။ Monitoring and Metric တွေ Logs တွေ ကြည့်ဖို့အတွက်လည်း AWS [CloudWatch](#) ရှိပါတယ်။ ကျွန်တော်တို့အနေနဲ့ Metric တွေကိုလည်း အမြဲမပြတ်ကြည့်နေဖို့ လိုပါတယ်။ ဘာလို့လဲဆိုတော့ Metric တွေက ရတဲ့ Data အချက်အလက်တွေက သင့်ကို Cost Optimization ပြု လုပ်ရာမှာ တစ်ထောင့်တစ်နေရာကနေ အထောက်အကူပြုမှာပါ။ အကြောင်းကိုနောက်မှဆက်ပြောပါမယ်။
- နောက်တစ်ဆင့်အနေနဲ့ ကျွန်တော်တို့ အနေနဲ့ Cost Optimization ပြု လုပ်မယ်ဆိုရင် ကိုယ်သုံးနေတဲ့ ဘယ် Service တွေကို လုပ်မယ်ဆိုတာ အရင် List လုပ်ရပါမယ်။ Cost တော့ လျော့ချချင်တယ် ဘယ်ဟာကိုလျော့လို့ လျော့ရမယ်မှန်းမသိလို့ မရပါဘူး။ ဘယ် Service တွေက သိပ်မသုံးဘူးလဲ EC2 Instance တွေကို မလိုပဲဖွင့်ထားမိလား ကိုယ်မသုံးပဲနဲ့ အပိုတွေ ဖြစ်နေတဲ့ Resource တွေရှိလား အရင်ကြည့်သင့်ပါတယ်။ အသုံးမလိုပဲဖွင့်ထားမိတဲ့ resource တွေက လွယ်လွယ်နဲ့ Cost Optimize လုပ်လို့ရတဲ့ အခြေခံအချက်ပဲဖြစ်ပါတယ်။ သဘောတရားကတော့ ရှင်းပါတယ်။ မသုံးပဲပိုနေတဲ့ Resource တွေကို ဖျက်ပစ်လိုက်ပါ။

အသုံးမပြုပဲ ပိုနေတဲ့ resource တွေကို ဖျက်တဲ့အခါမှာ မိမိ team member တွေနဲ့ သေချာ confirm လုပ်ပြီးမှဖျက်သင့်ပါတယ်။

- နောက်တစ်ခုကတော့ AWS က သူရဲ့ Ecosystem ထဲမှာ အသုံးပြုဖို့ ထုတ်ပေးထားတဲ့ Tools တွေကို အသုံးပြုပြီး ဘယ် Resource တွေက Utilization တွေနည်းလည်းဆိုတာတွေကို ကြည့်ပြီး Cost Optimize လုပ်ရပါမယ်။

Under Utilized ဖြစ်နေတဲ့ Resource တွေကို ကြည့်ဖို့ [AWS Trusted Advisor](#) နဲ့ Cost တွေရဲ့ Data Metric တွေကြည့်ဖို့ [AWS Cost Explorer](#) ရှိပါတယ်။

အကောင်းဆုံးက တော့ Tools တွေ အကူအညီ နဲ့ Monitoring က ရတဲ့ Metric Data တွေကို ကြည့်ပြီး ကိုယ်နဲ့ အကိုက်ညီဆုံးဖြစ်အောင် ကိုယ့် Organization နဲ့ Fit in ဖြစ်မဲ့ Cost Optimization ကိုပြုလုပ်လို့ရပါပြီ။

Right Sizing

Right Sizing ဆိုတာက တော့ ကိုယ်တော့ တို့ အနေနဲ့ ကိုယ့် ရဲ့ Infrastructure နဲ့ Application workloads တွေကို သေချာသိပြီဆိုရင်တော့ ဘယ် Instance တွေကတော့ Under Utilized ဖြစ်နေလဲ Over Provision ဖြစ်နေလဲဆိုတာကို ခွဲခြမ်းစိတ်ဖြာပြီး ရှိသင့် ရှိထိုက် ဖြစ်သင့် ဖြစ်ထိုက် သုံးသင့် သုံးထိုက် တဲ့ Resource

တွေကို သေချာရွေးချယ်လို့ပြောရပါမယ်။ ဆိုလိုတာက ဥပမာ Monitoring Metric တွေအရ CPU 2vCore RAM 4GB ပဲလို့တဲ့ Peak Hour တွေမှ အများဆုံးအဲသည်လောက်ပဲသုံးတဲ့ Application တင် run Instance Sizeကို ကျွန်တော်တို့က CPU 4vCore RAM 8GB ပေးထားမိတယ်ဆိုရင် ကျွန်တော်တို့က Over Provision ဖြစ်နေတယ်ဆိုတာသေချာပါပြီ။ အဲသည် Instance အတွက် Right Sizing ကိုမဖြစ်မနေလုပ်သင့်ပါပြီ။ Right Sizing ကျွန်တော်တို့က ပြုလုပ်ရာမှာလည်း Cost and Performance Trade-offs လေး တွေကိုလည်း သတိထားသင့်ပါတယ်။ Optimization လုပ်လိုက်လို့ Cost နည်းသွားပြီး Performance ကျသွားလို့ မရပါဘူး။ မရပါဘူးဆိုတာထပ် အဆင်မပြေတာပါ။ Performance ကျရင် Business Impact တွေ ရှိလာနိုင်ပါတယ်။ Infrastructure Price and Performance Trade-offs အတွက်လည်း AWS မှာဆိုရင် ကျွန်တော်တို့အနေနဲ့ Next Generation Resource တွေကို ရွေးချယ်အသုံးပြုခြင်းဖြင့် Cost လည်းအကုန်အကျသက်သာ Performance လည်းမထိခိုက် (တစ်ချို့ Resource တွေဆိုရင် Cost သက်သာပြီး Performance သိသိသာသာတိုးလာတာမျိုး) စေတဲ့ အတွက် တော်တော်လေးကိုအဆင်ပြေပါတယ်။

Latest Generation Resource ဆိုတာကတော့ AWS ရဲ့ EC2, RDS, ElastiCache, ElasticSearch မှာသုံးထားတဲ့ Underlying Hardware Resource တွေရဲ့ Generation ကိုဆိုလိုတာပါ။ ကျွန်တော်တို့ အနေနဲ့ Last Generation Resource တွေကနေ Price to performance ကို advantage ယူလို့ရပါတယ်။

A1 T3 T3a T2 M6g M5 M5a M5n **M4**

M4 instances provide a balance of compute, memory, and network resources, and it is a good choice for many applications.

Features:

- 2.3 GHz Intel Xeon® E5-2686 v4 (Broadwell) processors or 2.4 GHz Intel Xeon® E5-2676 v3 (Haswell) processors
- EBS-optimized by default at no additional cost
- Support for Enhanced Networking
- Balance of compute, memory, and network resources

Instance	vCPU*	Mem (GiB)	Storage	Dedicated EBS Bandwidth (Mbps)	Network Performance
m4.large	2	8	EBS-only	450	Moderate
m4.xlarge	4	16	EBS-only	750	High
m4.2xlarge	8	32	EBS-only	1,000	High
m4.4xlarge	16	64	EBS-only	2,000	High
m4.10xlarge	40	160	EBS-only	4,000	10 Gigabit
m4.16xlarge	64	256	EBS-only	10,000	25 Gigabit

All instances have the following specs:

- 2.4 GHz Intel Xeon E5-2676 v3** Processor
- [Intel AVX†](#), [Intel AVX2†](#), [Intel Turbo](#)
- [EBS Optimized](#)
- [Enhanced Networking†](#)

Previous Generation General Instance

A1	T3	T3a	T2	M6g	M5	M5a	M5n	M4
----	----	-----	----	-----	----	-----	-----	----

M5 Instances are the latest generation of General Purpose Instances powered by Intel Xeon® Platinum 8175 processors. This family provides a balance of compute, memory, and network resources, and is a good choice for many applications.

Features:

- Up to 3.1 GHz Intel Xeon® Platinum 8175 processors with new Intel Advanced Vector Extension (AVX-512) instruction set
- New larger instance size, m5.24xlarge, offering 96 vCPUs and 384 GiB of memory
- Up to 25 Gbps network bandwidth using Enhanced Networking
- Requires HVM AMIs that include drivers for ENA and NVMe
- Powered by the [AWS Nitro System](#), a combination of dedicated hardware and lightweight hypervisor
- Instance storage offered via EBS or NVMe SSDs that are physically attached to the host server
- With M5d instances, local NVMe-based SSDs are physically connected to the host server and provide block-level storage that is coupled to the lifetime of the M5 instance
- New 8xlarge and 16xlarge sizes now available.

Instance Size	vCPU	Memory (GiB)	Instance Storage (GiB)	Network Bandwidth (Gbps)	EBS Bandwidth (Mbps)
m5.large	2	8	EBS-Only	Up to 10	Up to 4,750
m5.xlarge	4	16	EBS-Only	Up to 10	Up to 4,750
m5.2xlarge	8	32	EBS-Only	Up to 10	Up to 4,750
m5.4xlarge	16	64	EBS-Only	Up to 10	4,750
m5.8xlarge	32	128	EBS Only	10	6,800
m5.12xlarge	48	192	EBS-Only	10	9,500


Latest Generation General Instance

အပေါ်က ပုံ ၂ ပုံမှာတော့ Previous Generation General Instance (M4) နဲ့ Latest Generation General Instance (M5) Performance ကွာခြားပုံလေးကိုပြထားတာဖြစ်ပါတယ်။ Latest Generation မှာ ဘာတွေ Advantage ရလဲဆိုရင် သိသိသာသာဖြစ်တာက Underlying CPU Generation , Network Bandwidth (Gbps) နဲ့ EBS Bandwidth (Mbps) တို့ပဲဖြစ်ပါတယ်။ ဒါကတော့ Performance Advantage ပါ။

နောက်တစ်ခုက Latest Generation Instance တွေရဲ့ **Cost Advantage** ပါ။







Choose region: Asia Pacific (Singapore) ▾

Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

 Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances.

Clear Form

Compute: Amazon EC2 Instances:

	Description	Instances	Usage	Type	Billing Option	Monthly Cost
		1	100 % Utilized/Mo ▾	Linux on m4.large	 On-Demand (No Contrac 	\$ 91.50
		1	100 % Utilized/Mo ▾	Linux on m5.large	 On-Demand (No Contrac 	\$ 87.84

M4 vs M5 Cost Comparison

Choose region: Asia Pacific (Singapore) Inbound Data Transfer is Free and Outbound Data Transfer is 1 GB free per region per month

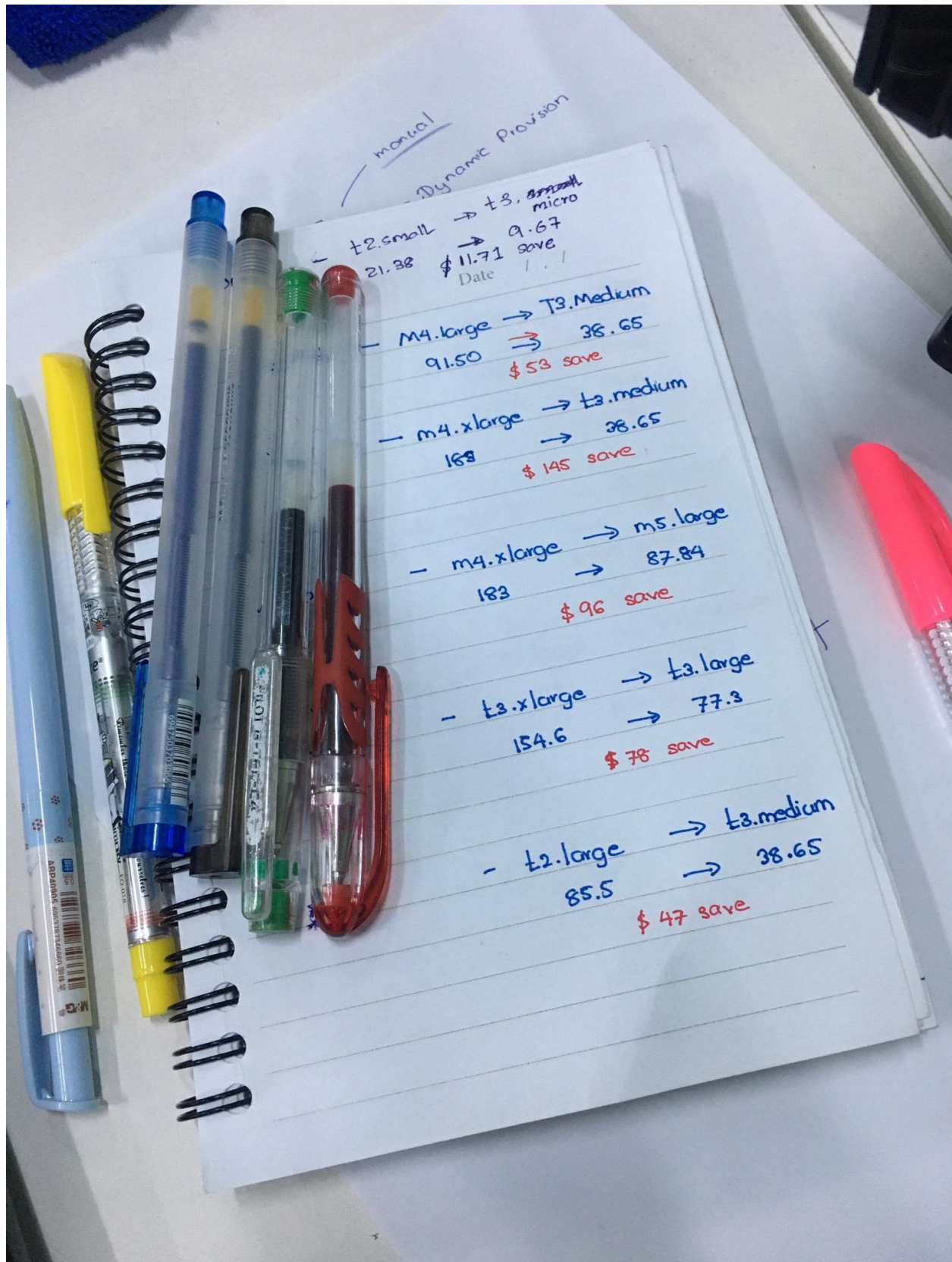
Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers. Amazon Elastic Block Store (EBS) provides persistent storage to Amazon EC2 instances. Clear Form

Compute: Amazon EC2 Instances:

	Description	Instances	Usage	Type	Billing Option	Monthly Cost
		1	100 % Utilized/Mo	Linux on m4.large	On-Demand (No Contrac	\$ 91.50
		1	100 % Utilized/Mo	Linux on m5.large	On-Demand (No Contrac	\$ 87.84
		1	100 % Utilized/Mo	Linux on c4.large	On-Demand (No Contrac	\$ 84.18
		1	100 % Utilized/Mo	Linux on c5.large	On-Demand (No Contrac	\$ 71.74
		1	100 % Utilized/Mo	Linux on t2.xlarge	On-Demand (No Contrac	\$ 342.00
		1	100 % Utilized/Mo	Linux on t3.2xlarge	On-Demand (No Contrac	\$ 309.20

Previous Generation vs Latest Generation Cost Comparison

M4 vs M5 , C4 vs C5 , T2 vs T3 ကိုကြည့်မယ်ဆိုရင် Latest Generation ကိုသုံးခြင်းဖြင့် Cost and Performance Advantage ရမှာပါ။ အချုပ်ကိုပြောရမယ်ဆိုရင်တော့ Right Sizing လုပ်တဲ့ အခါ ဒီလိုအချက်တွေထည့်သွင်းစဉ်းစားဖို့လိုပြီး Cost and Performance Advantage ကိုယူကာ Over Provision နဲ့ Under Provision ဖြစ်နေတဲ့ Resource တွေကို Optimize ပြုလုပ်နိုင်မှာပဲဖြစ်ပါတယ်။



Right Sizing Example

ဒီအပေါ်က ပုံကတော့ စာရေးသူလက်ရှိလုပ်ကိုင် နေတဲ့ Company မှာ Cloud Team နဲ့ Software Architect Team, Development Team တို့ အတူတကွ တိုင်ပင်ပြီး Cost Optimize လုပ်ထားတဲ့ ပုံပဲဖြစ်ပါတယ်။

ဒီလောက်ဆိုရင်တော့ AWS Cloud မှာ Cost Calculation , Billing အကြောင်းနဲ့ Cost Optimization အကြောင်းကို အနည်းနဲ့ အများနားလည်မယ်လို့ ယုံကြည်ပါတယ်။ Cost Optimization နဲ့ ပတ်သက်ပြီး နောက်ထပ် Article တွေဆက်ရေးသွားဖို့ရှိလို့ ဆက်လက်အားပေးကြပါဦး ခင်ဗျာ :D

Cost Optimization on AWS Cloud - Part 2

Overview of AWS Services' Pricing and Cost Optimization

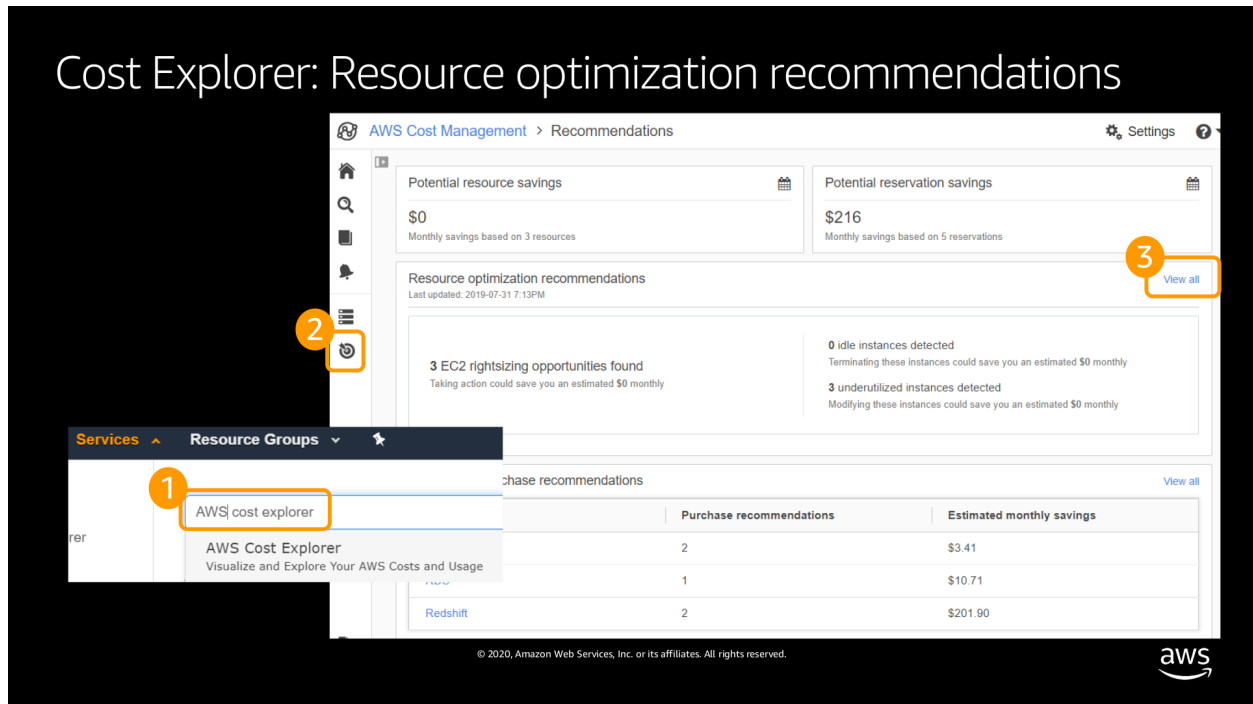
ကျွန်တော်တို့ အရင် [article](#) မှာတော့ Cost Optimize လုပ်ဖို့လိုအပ်တဲ့ အရာလေးတွေနဲ့ Billing သဘောတရားများကို အကြမ်းဖျင်းအနေနဲ့ ပြောပြီးပြီဆိုတော့ ဒီ Part 2 မှာနောက်ထပ် အကြောင်းအရာလေးတွေ ကိုထပ်ပြီးဆွေးနွေးကြရအောင်ဗျာ။

Cost Optimization နဲ့ ပတ်သက်တဲ့ ဆက်စပ်နေတဲ့ နောက်ထပ်အကြောင်း အရာတစ်ခုက တော့ Right Sizing ပါ။ Right Sizing ပြုလုပ်တာက ကိုယ့် ရဲ့ infrastructure cost ကို သိသိသာသာသက်သာစေပါတယ်။

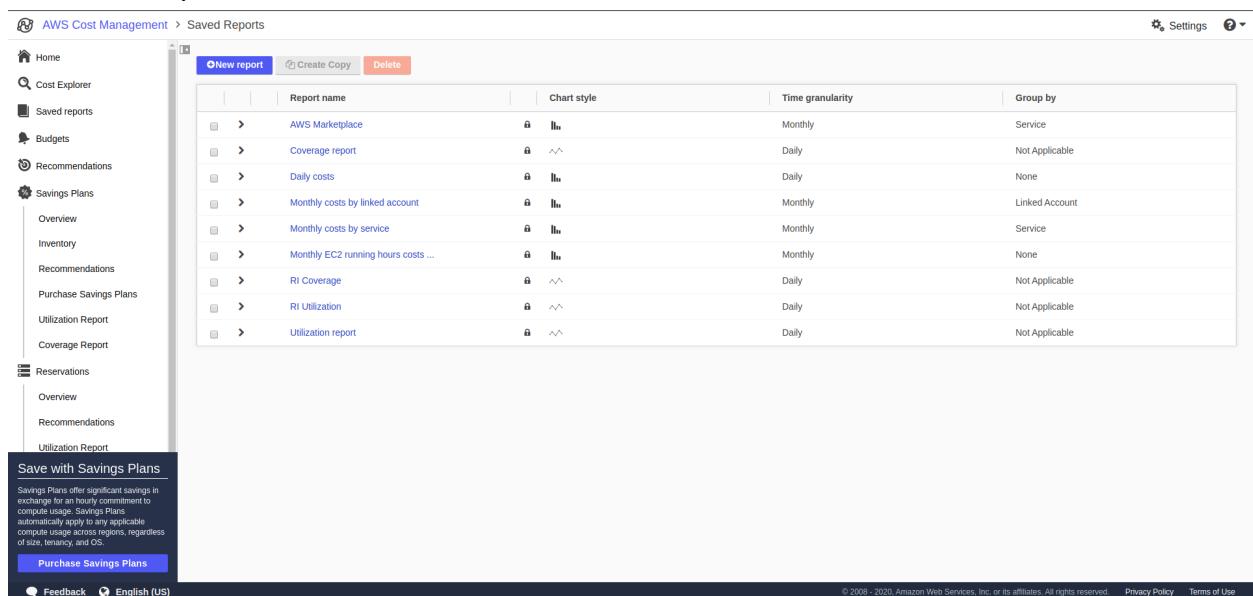
AWS မှာ Right Sizing လုပ်မယ်ဆိုရင်တော့ ကျွန်တော်တို့ အနေနဲ့ ရှေ့ article မှာပြောခဲ့သလိုတွေပြုလုပ်နိုင်ပါတယ်။ နောက်ပြီးတော့ ကျွန်တော်တို့ အနေနဲ့ AWS က သူ Ecosystem မှာ Right Sizing ပြုလုပ်နိုင်ဖို့ ထုတ်ပေးထားတဲ့ Tools တွေကို သုံးပြီးလည်းပြုလုပ်နိုင်ပါတယ်။

- ပထမဆုံးတစ်ခုက **AWS Cost Explorer** ပါ။ ဒီကောင်လေးကတော့ ကိုယ်သုံးနေတဲ့ ဘယ် resource တွေက ဘယ်လောက်တော့ ကုန်ကျနေတယ်ဆိုတာကိုပြန်ကြည့်လို့ရပါတယ်။ နောက်ပြီး လွန်ခဲ့တဲ့ ၇ရက် ၁၀ ၂၀ ကကိုယ်သုံးခဲ့တဲ့ resource တွေရဲ့ utilization ကိုအခြေခံပြီး ဘယ် resource တွေကတော့ reserved လုပ်သင့်တယ် တစ်ချို့ဆိုလည်း Down Sizing လုပ်သင့်တယ် ဘာညာစတာတွေကို ပြပေးထားတဲ့ အတွက် ကျွန်တော်တို့ က သူ tools က ထုတ်ပေးတဲ့ Data တွေနဲ့ ကိုယ့် environment , organization ကလိုချင်တဲ့ လိုအပ်တဲ့အရာတွေနဲ့ ကိုက်ညီတယ်ဆိုရင် ကျွန်တော်တို့က လွယ်လွယ်ကူကူပဲ Cost Optimize ပြုလုပ်နိုင်မှာပါ။

Cost Explorer: Resource optimization recommendations



AWS Cost Explorer



AWS Cost Explorer

- နောက်ထပ်တစ်ခုကတော့ AWS Trusted Advisor ပါ။ သူကတော့ ကိုယ့် AWS account ရဲ့ Support Plan ပေါ်မူတည်ပြီး သုံးလိုရတဲ့ Features တွေတော့ ကွာခြားချက်ရှိပါတယ်။ ကိုယ်တော်တို့ ကတော့ Business Plan ယူထားတဲ့ အတွက် Feature အစုံသုံးလို့ရနေပါတယ်။ Trusted Advisor ထဲက Cost Optimization Checks ဆိုတဲ့ feature ရဲ့ အကူအညီနဲ့လည်း Low utilization EC2 Instance တွေ Idle ဖြစ်နေတဲ့ RDS Instance တွေ မသုံးပဲ ထားမိနေတဲ့ Resources တွေကို ပြပေးပါတယ်။ အဲ့သည်လိုပြပေးတဲ့ အတွက်ကျွန်တော်တို့က ဘယ်

resource တွေကကိုယ်တစ်ကယ်မသုံးပဲ ပိုက်ဆံကုန်နေသလဲ ဆိုတာကို မြန်မြန်ဆန်ဆန်သိနိုင်တာပေါ့ဗျာ။

Dashboard

Cost Optimization

Performance

Security


Fault Tolerance

Service Limits


Preferences

AWS Trusted Advisor


Find unused resources using tools like AWS Trusted Advisor (available via business support)




AWS Trusted Advisor




Old snapshots




Unattached Elastic IPs



Idle EC2, RDS instances




Unattached EBS



Idle or unattached load balancers

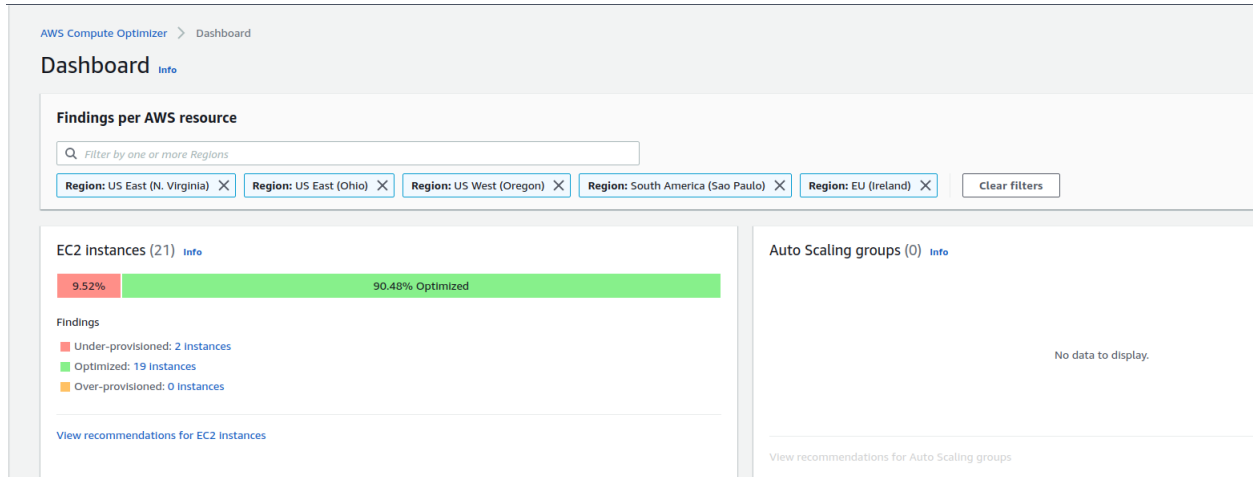
© 2020, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Trusted Advisor

- နောက်ထပ်တစ်ခုကတော့ AWS Compute Optimizer ဆိုတဲ့ Tools လေးပါ။ သူကတော့ AWS ရဲ့ Service တစ်ခုအနေနဲ့ဖြစ်လာတာတော့ မကြာသေးပါဘူး။ ဒါပေမယ့် တော်တော်လေးအသုံးဝင်တဲ့ Tool တစ်ခုပါ။ Compute Optimizer ဆိုတဲ့ အတိုင်းပဲ EC2 Instance တွေ Optimize ဖြစ်ရဲ့လား Over Provision/Under Provision

ဖြစ်နေလားဆိုတာကိုကြည့်လို့ရမယ်၊ မြန်မြန်ကြည့် မြန်မြန်သိတော့ Cost Optimize မြန်မြန်လုပ်နိုင်တာပေါ့ဗျာ။ လက်ရှိမှာတော့ Compute Resource တွေထဲကမှာ AWS EC2 Instance နဲ့ Autoscaling Group တွေအတွက်ပဲ အသုံးပြုလို့ရနိုင်ပါသေးတယ်။ နောက်ထပ်တစ်ခုက အခုအချိန်မှာတော့ AWS Region ၅ ခုမှာပဲ အသုံးပြုနိုင်ပါသေးတယ်။



AWS Compute Optimizer

ကျွန်တော့်ကိုယ်ပိုင် အမြင်နဲ့ပြောရရင်တော့ ဒီ tools လေးတွေကတော်တော် အထောက်အကူပြုပါတယ်။ ကျွန်တော်တို့အနေနဲ့ Instance တစ်လုံး နှစ်လုံးဆိုရင်တော့ tools တွေ အကူအညီမယူပဲ ကိုယ့်ဘာသာကိုယ်တွက်လို့ရနိုင်ပါတယ်။ Server အလုံး တစ်ဆယ် နှစ်ဆယ် တစ်ရာ နှစ်ရာဖြစ်လာပြီဆိုရင်တော့ ဒီ tools တွေရဲ့ အကူအညီနဲ့ Monitoring Metric ကရတဲ့ဒေတာတွေနဲ့ ပေါင်းပြီး မှန်မှန်ကန်ကန်နဲ့ မြန်မြန်ဆန်ဆန် Cost Optimized ပြုလုပ်နိုင်မှာဖြစ်ပါတယ်။

Reserved Instance

နောက်ထပ် Cost Optimize လုပ်နိုင်မဲ့ အချက်ကတော့ မိမိ Infrastructure က Steady Workload လိုမျိုးရှိနေပြီဆိုရင် ကိုယ့်ရဲ့ Business နဲ့လည်းကိုက်ညီမယ်ဆိုရင် Reserved လုပ်ခြင်းကို ရွေးချယ် ။ ဥပမာ ကျွန်တော်တို့ Gym မှာပိုက်ဆံပေးပြီးသွားဆော့သလိုပဲ တစ်နေ့ဆော့ရင် (On-Demand) ဆိုရင် ဈေးတစ်မျိုး ဘန်စာကြိုပေးပြီးဆော့မယ်ဆိုရင်ဈေးတစ်မျိုး (AllUpFront) ဘ နှစ်တော့ဆော့မယ် တစ်လချင်တစ်မျိုး (NoUpfront) ဘနှစ်တော့ ဆော့မယ် ၆လတစ်ဖြတ်ပေးမယ် (PartialUpFront) ဆိုတဲ့ သဘောတရားပါပဲ။ ဆိုလိုတာက ကိုယ့်ရဲ့ Infra အနေအထား မြေစားက ပြောင်းဖို့မရှိဘူး ဒါမှမဟုတ် တစ်နှစ်အတွင်းဒီ ဟာကို အပြီပြင်သုံးမယ်ဆို Reserved Instance က On-Demand ထက်ပို ဈေးသက်သာပါတယ်။ ဆိုတော့ Compute EC2 Reserved Instance သုံးတော့မယ်ဆို သိသင့်တာလေးတစ်ခုက သူရဲ့ အမျိုးအစားလေးပါပဲ ဘာတွေလဲဆိုတော့

1. **Standard**
2. **Convertible**

တွေပဲဖြစ်ပါတယ်။ အဲ့ထဲမှာ Standard က Convertible ထက်ပိုပြီးဈေးကိုသက်သာစေပါတယ်။ အဲ့နှစ်မျိုးဘာတွေကွာလဲ ဆို တော့ Standard က သာမန် တစ်နှစ် သုံးနှစ် Reserved လုပ်လိုက်ပြီဆိုရင်

သုက Region or AZ ပြောင်းလိုတော့ရပါတယ် နောက်ပြီး Instance Flexible ရတယ်ဆို တော့ Same Instance Family က Size အတိုးလျော့လုပ်လို့ရပါတယ် ဒါပေမဲ့ Instance Family တော့ပြောင်းမရပါဘူး ဆိုလိုတာ က T-series standard reserved ကို တစ်နှစ်ဝယ်ထားရင် M-series ကို ပြောင်းမရပါဘူး။ ဒါလေးက Standard Reserved ရဲ့ Limitation လေးပေါ့။ Convertible ဆိုရင်တော့ Standard ထက်နည်းနည်းလေးပိုချေးပေးရမယ် ဒါပေမဲ့ သုက convertible ဆိုတဲ့အတိုင်း Instance Size or Family လေးတွေပြောင်းလို့ရတယ်။ အကျယ်တဝန် ကိုလည်း သုရဲ့ Official Documentation မှာဝင်ရောက်ဖတ်လို့ရပါတယ်။ ဒါဆို reserved instance ကို ယေဘုယျ တော့ သဘောပေါက်လောက်ပြီထင်ပါတယ်။ ခြုံပြောရရင်တော့ ကျွန်တော်တို့ က အချိန်ဘယ်လောက်အတိုင်းအတာသုံးပါမယ်ဆိုပြီး commit လုပ်ထားမယ် အချိန်များများ commit လုပ်ရင် Discount (31%~75%) လောက်ထိ များများသက်သာမယ်ပေါ့ဗျာ။

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/reserved-instances-types.html>

<https://aws.amazon.com/blogs/aws/new-instance-size-flexibility-for-ec2-reserved-instances/>

Reserved ဝယ်တဲ့ အခါမှာ ကိုယ်တစ်ကယ်သုံးချင်တဲ့ Region နဲ့ တစ်ကယ်သုံးခင်တဲ့ လိုအပ်တဲ့ အရေအတွက် ကိုပဲဝယ်ယူသင့်ပါတယ်။ ပိုဝယ်မိလိုက်တာတို့ ဘာတို့ဆိုရင် ဒီကုန်ကျစရိတ်ကို ကိုယ်ကမသုံးလည်းပဲ ပေးချေနေရတော့မှာပါ။

Characteristic	Standard	Convertible
Terms (avg. discount off On-Demand)	1yr (40%), 3yr (60%)	1yr (31%), 3yr (54%)
Change Availability Zone, instance size (for Linux OS), networking type	Yes (Using ModifyReservedInstances API and console)	Yes (Using ExchangeReservedInstances API and console)
Change instance families, operating system, tenancy, and payment option		Yes
Benefit from Price Reductions		Yes

RI Standard Convertible

Elasticity

Cost Optimize ပြုလုပ်နိုင်မဲ့ နောက်တစ်ချက်အနေနဲ့ကတော့ Elasticity ပဲဖြစ်ပါတယ်။

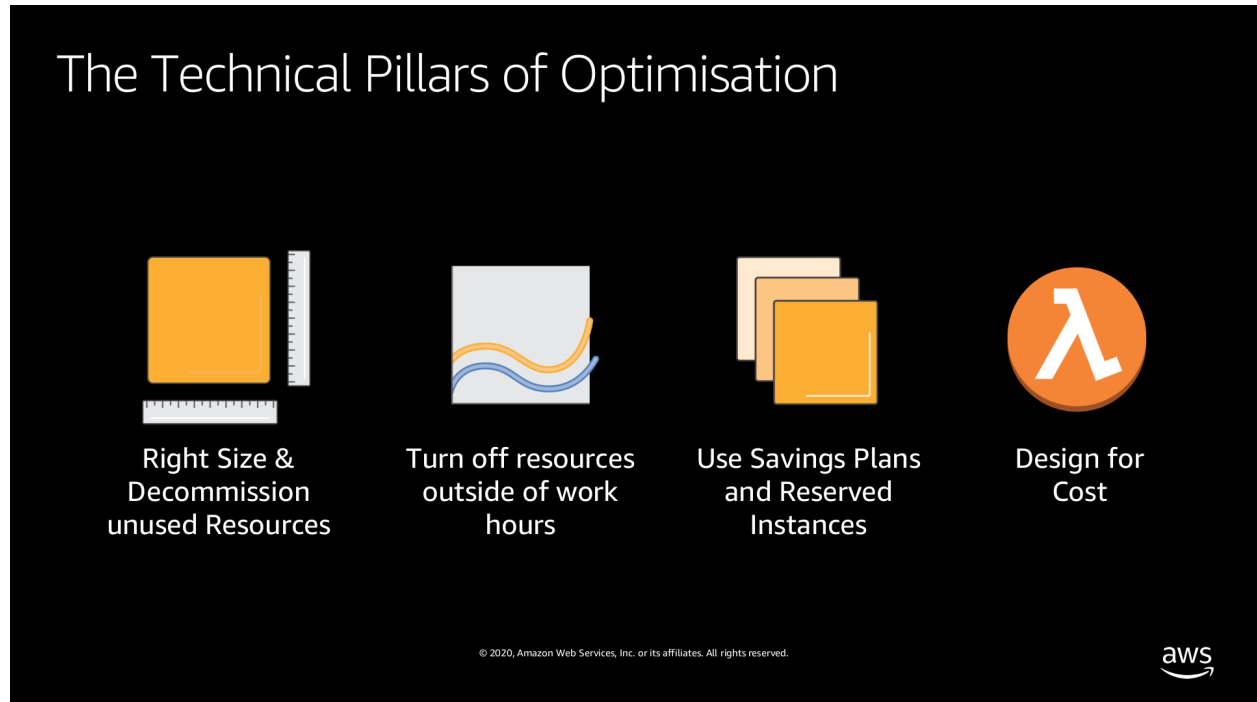
Elasticity အနေနဲ့ဆိုရင် အဓိက ၂ ပိုင်းခွဲလို့ရတယ်ပေါ့နော်

1. **Time-Based**
2. **Volume-Based**

Time-Based

မှာသိသိသာသာ Cost လျှော့နိုင်တာကတော့ ကျွန်တော်တို့ရဲ့ **Development/QA/Staging Environment က Server တွေကို ကိုယ့်ရုံးချိန်အလိုက် အပိတ်အဖွင့်ပြုလုပ်တာမျိုးပေါ့။** AWS EC2 Instance တွေက EBS ကလွဲရင် ပိတ်ထားရင် ပိုက်ဆံမကောက်ပါဘူး။ အဲသည်အတွက် ရုံးချိန်ပြင်ပနဲ့ စနေ တနင်္ဂနွေလိုနေ့မျိုးတွေမှာ ကိုယ်ရဲ့ Development/QA/Staging environment က Instance တွေကို EC2 Instance Scheduler နဲ့ Auto **START/STOP** လုပ်ထားမယ်ဆိုရင်ရုံးချိန်အတွင်းသုံးတဲ့ စာရီစာလောက်ပဲကုန်ကျမှာဖြစ်ပါတယ်။

Please don't ever try on your production environment. If yes, do as your own RISK !!



Turn of resources

Volume-Based

ဥပမာ ကျွန်တော်တို့ရဲ့ Web Server က LoadBalancer အနောက်မှာ အမြဲတမ်း ၂ လုံး Up and Running ဖြစ်နေရမယ်ဆိုပါစို့။ အဲသည်မှာ ကျွန်တော်တို့ Company က Promotion ချလို့ User Base ကအရမ်းများလာပြီဆိုရင် အနောက်က web server ၂ လုံးက Load မခံနိုင်တော့ဘူးဆိုရင် ကျွန်တော်တို့ ကလုပ်စရာ ၃ မျိုးလောက်ဖြစ်လာပြီ။ အဖြေရှာရင် ကျွန်တော်တို့က User တွေအသုံးမများခင်မှာ ကြိုတင်ပြီးတော့ server ကို instance size ကြီးကြီးသုံးထားမလား ၊ အဲလိုသုံးလိုက်ပြန်ရင်လည်း User တွေမသုံးခင် ကြိုပြင်ထားရဆိုတော့ Cost effective လည်းမဖြစ်လှဘူး မသုံးပဲကြိုပေးနေရမှာ၊ နောက်တစ်ခုက User တွေများလာတော့မှ Server ကို Vertically Scaling - CPU, Memory တွေထပ်တိုတာမျိုး လုပ်မလား၊ သည်လိုဆိုရင်လည်း အဆင်မပြေသေးပြန်ဘူး။ အဲသည်မှာ Elasticity ဆိုတာလိုလားတယ်။ ဒီနေရာမှာ Customer Demand ဘယ်လောက်များများ ကျွန်တော်တို့ Infrastructure က အလိုလလျောက် ကြိုနှိုင်းဆံ့နိုင်မယ်ဆိုရင် Business လည်းမထိခိုက်ဘူး Customer တွေရဲ့ စိတ်ကျေနပ်မှုကလည်း တိုးလာရင်တော့ မကျသွားတော့ဘူးဗျာ။ နောက်တစ်ခုက အဓိက အချက်ဖြစ်တဲ့ Cost Effective လည်းဖြစ်မယ်ဆိုရင် Win-win ပဲ။ သည်တော့ ကျွန်တော်တို့ **AWS Auto**

Scaling လိုဟာမျိုးကိုသုံးပြီး Customer Demand နဲ့ကိုက်ညီအောင် Auto Scale Out / In လုပ်လို့ရနိုင်မယ်။ ဆိုလိုတာကတော့ အသုံးပြုတဲ့ User တွေများလာလို့ CPU Utilization , Network Utilization , Memory Utilization ဘယ်လောက်ရောက်ရင်တော့ အစက Server ၂ လုံးရှိတာကနေ နောက်ထပ် ၃ လုံးလား ၅လုံးလာ Auto Scale Out လုပ်လို့ရနိုင်မယ်။ အိုကေ အဲဒါမှာ User တွေသိပ်မသုံးတော့ဘူး ဆိုလည်း Server ရဲ့ resource utilization လည်းပြန်ကျသွားရင် ကျွန်တော်တို့က Auto Scale In ဖြစ်အောင်လုပ်ထားမယ် Elasticity ရှိမယ်ဆိုရင် အပေါ်မှာပြောခဲ့သလိုပဲ Win-win solution ဖြစ်မယ်ပေါ့ဗျာ :)



Auto Scaling

Spot Instance

Spot Instance အသုံးပြုခြင်းဟာလည်း Cost Optimization လုပ်တဲ့နေရာမှာ အရေးပါပါတယ်။ Spot Instance တွေက ကျွန်တော်တို့သုံးနေတဲ့ On-Demand Instance တွေနဲ့ ဈေးနှုန်းခြင်းယှဉ်ရင် 60% ကနေ 90% လောက်ထိကိုသက်သာပါတယ်။ Spot Instance ကတော့ AWS ကသူရဲ့ ပိုလျှံနေတဲ့ spare compute resource တွေကို ဈေးသက်သက်သာသာနဲ့ ငှားတဲ့ သဘောပါပဲ။ Spot Instance အကြောင်းကို **ဒီမှာ** လည်းဖတ်ရှုလို့ရပါတယ် ခင်ဗျာ။

Savings Summary

A high-level summary of your savings across all of your **running** and **recently terminated** Spot Instances.

For detailed reporting on your account-level Spot usage, visit [Cost Explorer](#).

Spot usage and savings

4	474	1770	\$26.47	\$7.83	70%
Spot Instances	vCPU-hours	Mem(GiB)-hours	On-Demand total	Spot total	Savings

Average cost per vCPU-hour: **\$0.0165**
Average cost per mem(GiB)-hour: **\$0.0044**

Details

m5.large (3)	432 vCPU-hours	1728 mem(GiB)-hours	\$7.66 total	70% savings
t3.small (1)	42 vCPU-hours	42 mem(GiB)-hours	\$0.17 total	70% savings

* Spot savings are estimated savings and may differ from actual savings. This is because the savings shown on this page do not include the billing adjustments for your usage.

Close

Spot Instance cost saving

သည်လောက်ဆိုရင် တော့ မိမိတို့ Organization နဲ့ကိုက်ညီမှု Cost Optimization ကိုအလွယ်တကူပြုလုပ်နိုင်မယ်လို့ မျှော်လင့်မိပါတယ်။ နောက်ထပ် နောက်ထပ် အကြောင်းအရာတွေနဲ့ပတ်သက်ပြီး ရေးသား ဝေမျှသွားဦးမှာဖြစ်တဲ့ အတွက်ဆက်လက်အားပေးကြပါဦးခင်ဗျာ။