From AZs to the Internet: Cracking the code on AWS networking costs

Akshay Kapoor

Cloud Infrastructure @ AWS

15+ Years

Consult AWS customers

(since ~5 years)

Domains

Software Development

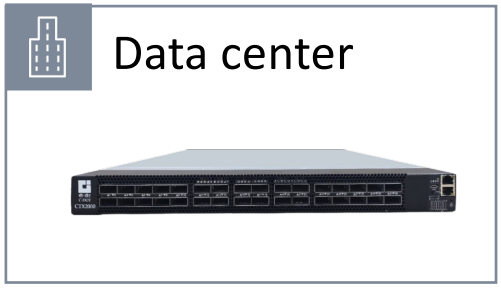
DevOps

Cloud Infrastructure

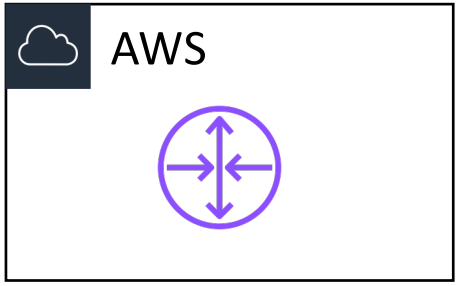
Book Author

AWS DevOps Simplified

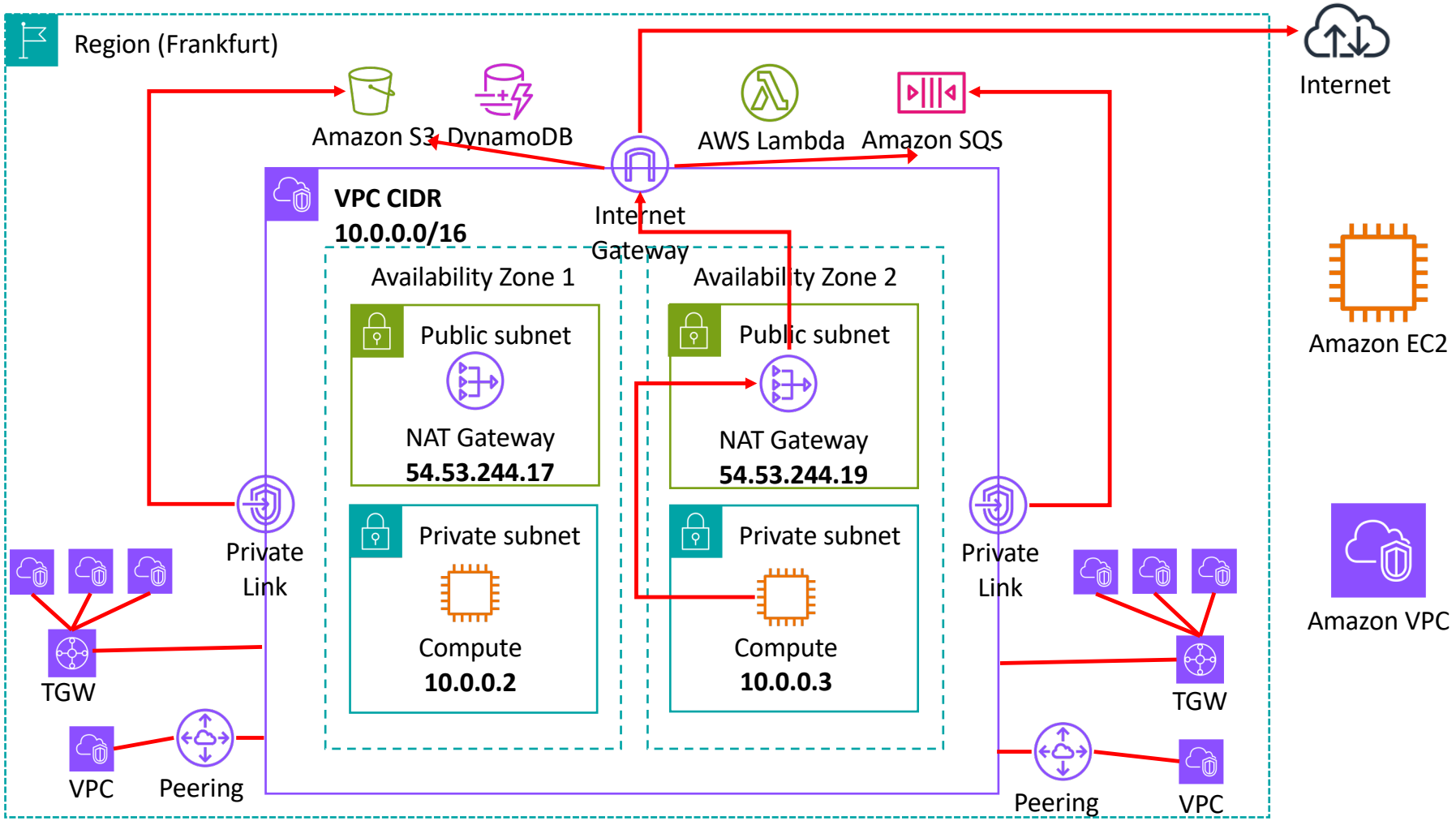
Networking in cloud is peculiar



v/s



Where do you host your workloads?


Region (Frankfurt)
Internet
Amazon S3
DynamoDB
AWS Lambda
Amazon SQS
VPC CIDR
10.0.0.0/16
Internet Gateway
Availability Zone 1
Availability Zone 2
Public subnet
NAT Gateway
54.53.244.17
Public subnet
NAT Gateway
54.53.244.19
Private subnet
Compute
10.0.0.2
Private subnet
Compute
10.0.0.3
Private Link
Private Link
TGW
TGW
VPC
Peering
Peering
VPC
Amazon EC2
Amazon VPC

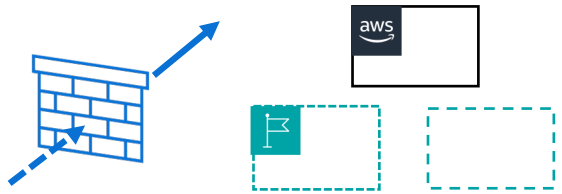
Let’s simplify the cost peculiarity

Dimensions of AWS Networking costs

Data Transfer Charges

*How much data crosses an **infrastructure boundary***

$ / GB



More nuanced than others

Service Charges

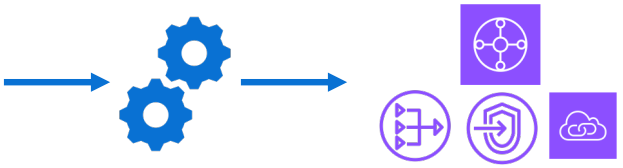
*How long is a **networking service** running for*

$ / hr

Data Processing Charges

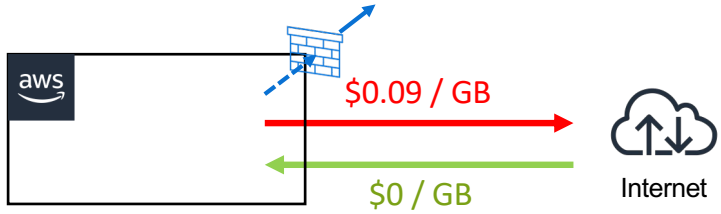
*How much data is processed by a **networking service***

$ / GB

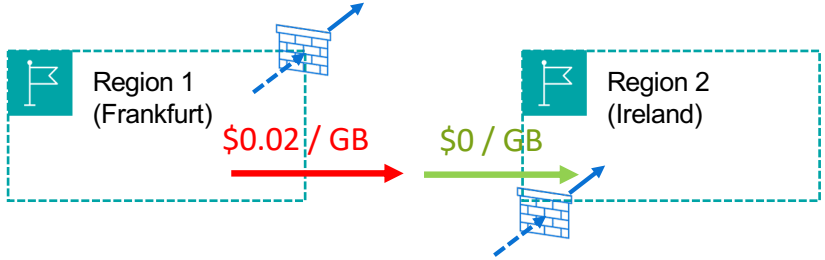


Data transfer – be wary of crossing the "boundaries"

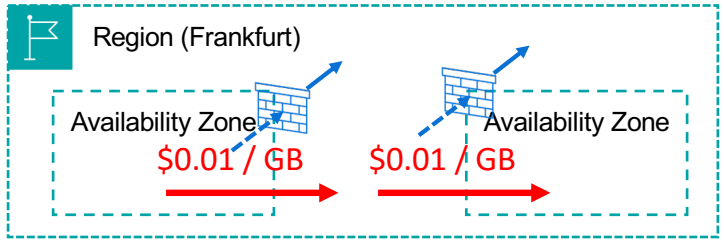
AWS(Region) -> Internet



Region 1 -> Region 2

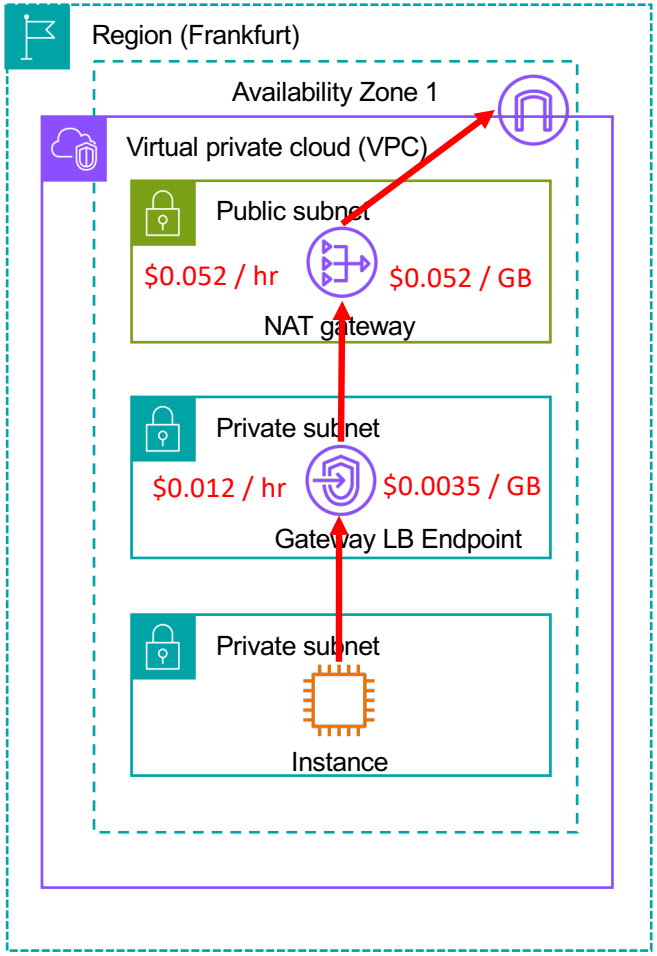


AZ-1 -> AZ-2



*Note: All costs are for **eu-central-1** region (Frankfurt). Might vary for others*

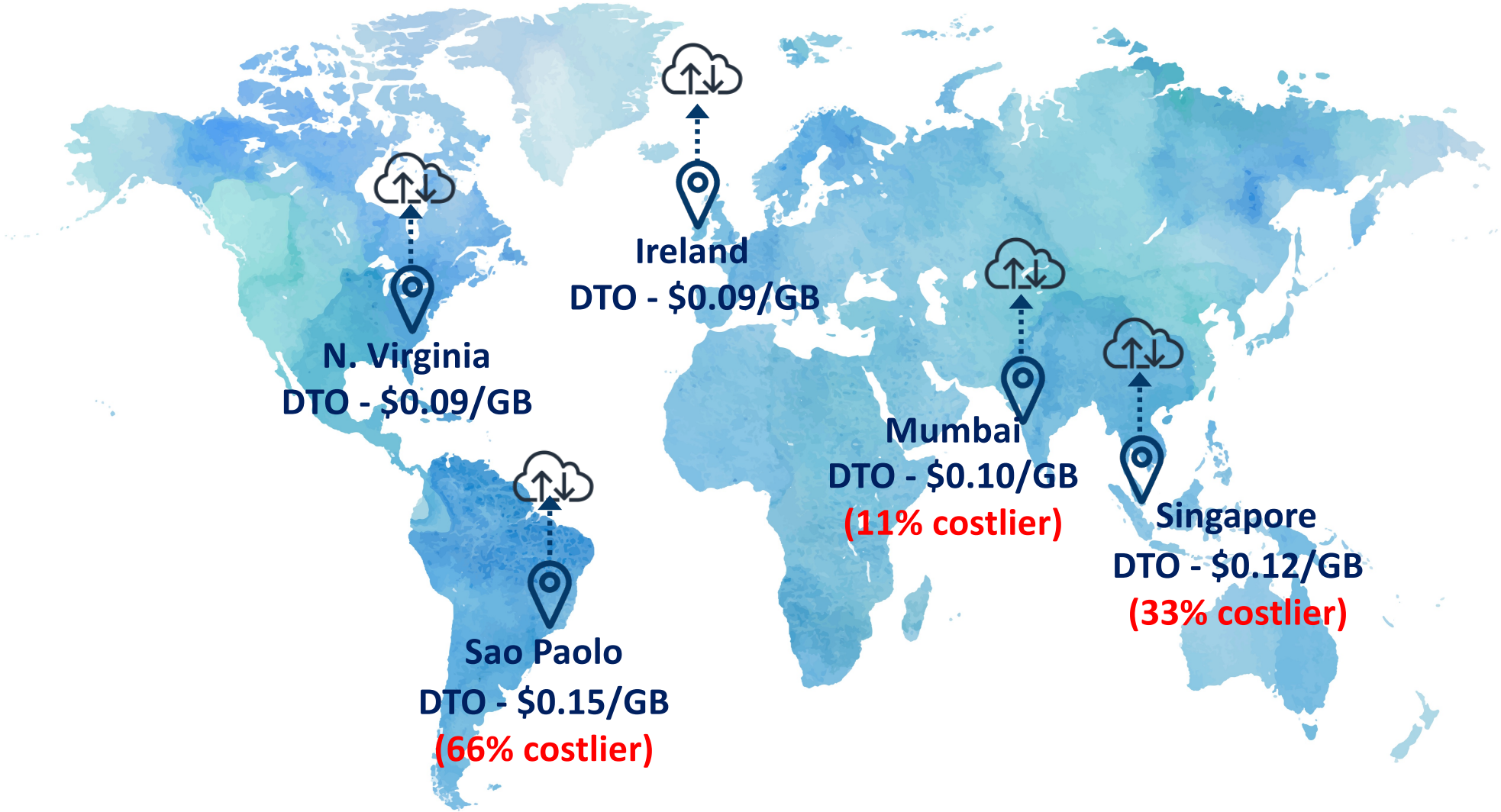
Data processing + Service Cost (NAT Gateway + GWLB)



…back to the customer case

210,000
USD/month

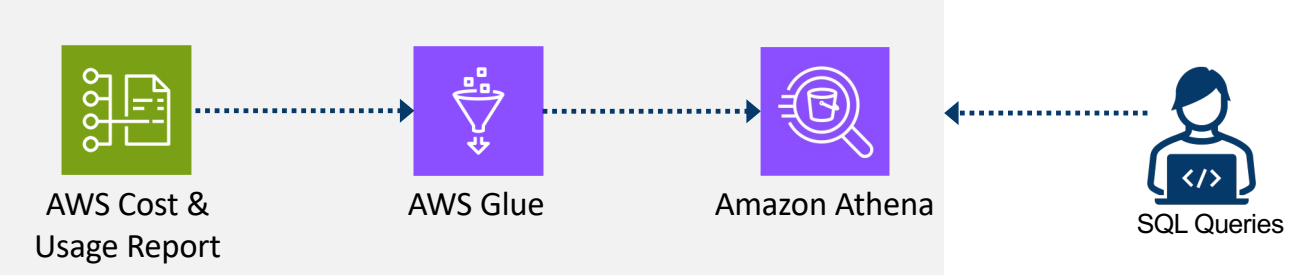
Which AWS regions are used?


Ireland
DTO - $0.09/GB
N. Virginia
DTO - $0.09/GB
Mumbai
DTO - $0.10/GB
(11% costlier)
Singapore
DTO - $0.12/GB
(33% costlier)
Sao Paolo
DTO - $0.15/GB
(66% costlier)

The **AWS region(s)** you choose to host your Workloads will **influence your data transfer costs**!

Dissecting data transfer costs

Data Transfer Costs



	Charge Type	Cost
1	IntraRegion	102,281.53
2	InterRegion Outbound	37,093.13
3	AWS Outbound	17,766.04
4	Inter Region Peering Data Transfer Outbound	3,619.92
5	CloudFront to Origin	9.79456E-5
6	IntraRegion-xAZ-In	0.0

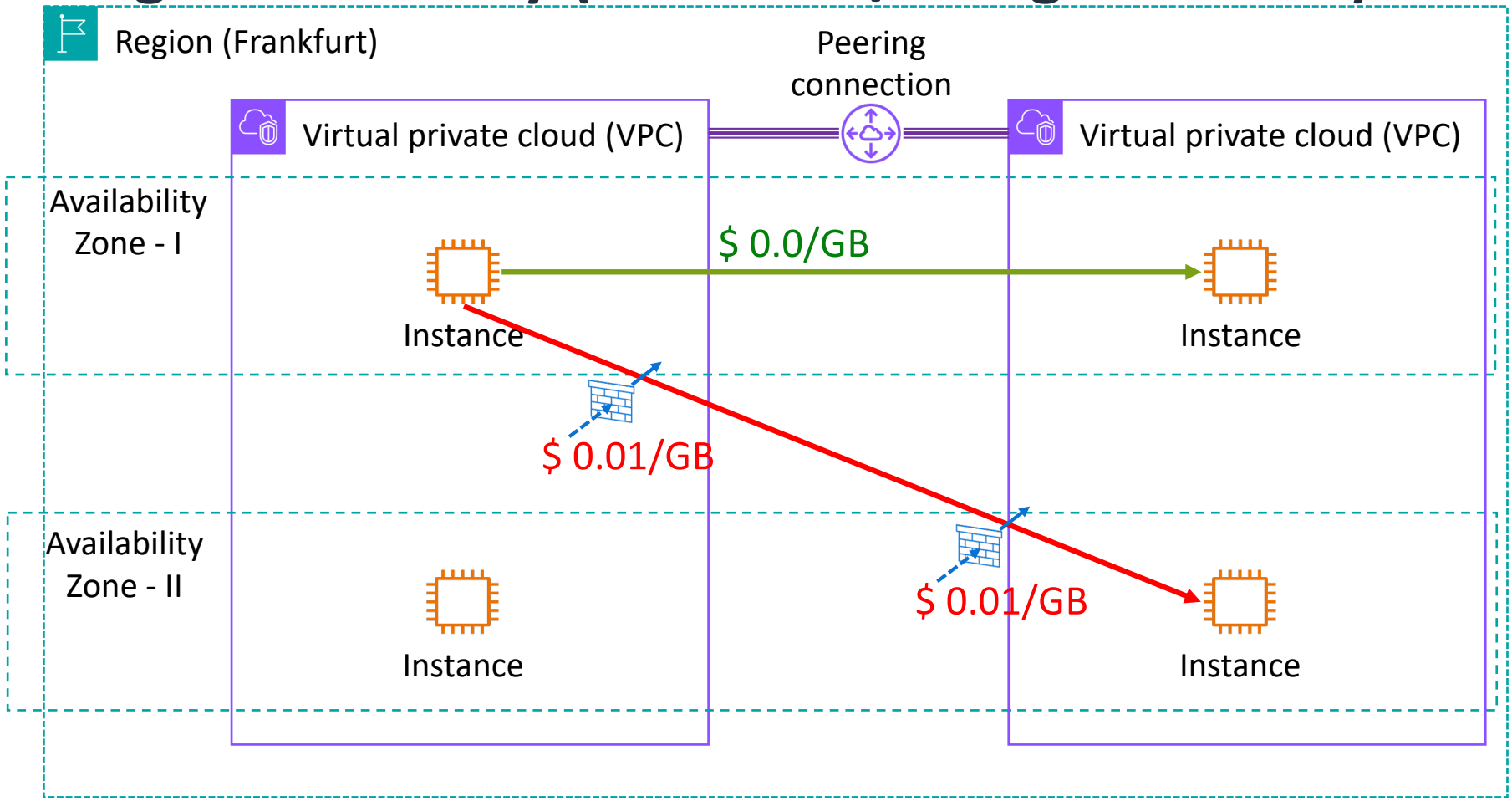
Data transfer costs
(Intra-Region)

1	IntraRegion	102,281.53
2	InterRegion Outbound	37,093.13
3	AWS Outbound	17,766.04
4	Inter Region Peering Data Transfer Outbound	3,619.92
5	CloudFront to Origin	9.79456E-5
6	IntraRegion-xAZ-In	0.0

Dissecting the “*Intra-Region*” costs

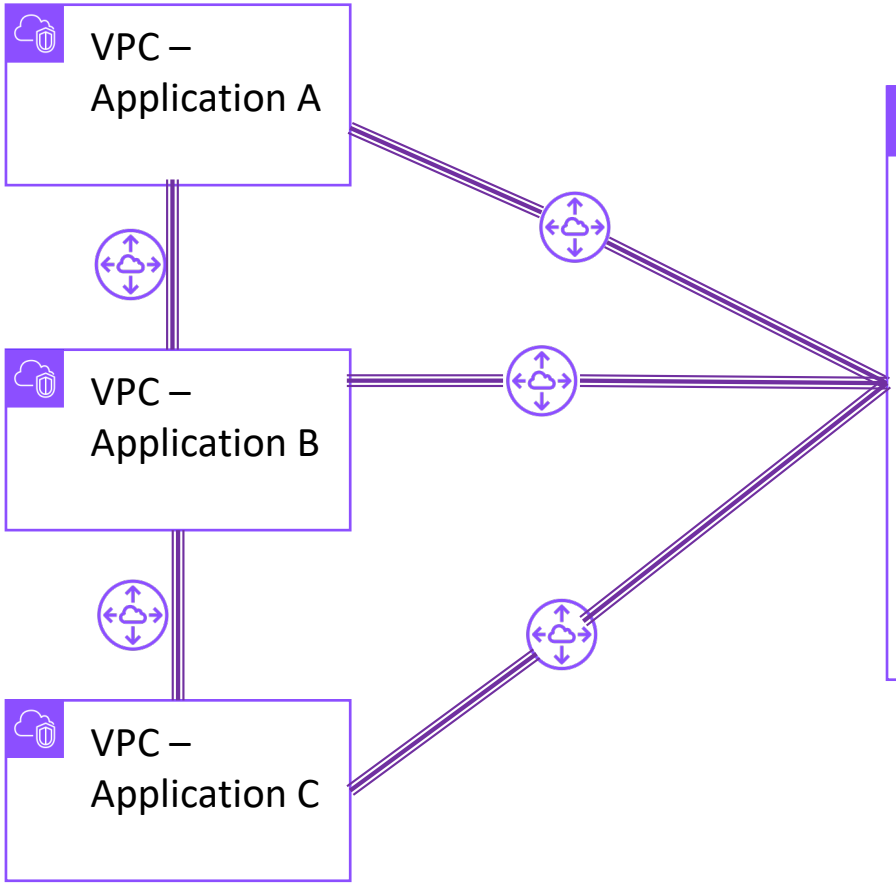
	Cost	Product Code	Charge Type
1	40,378.47	AmazonEC2	EU-DataTransfer-Regional-Bytes InterZone-In
2	26,596.16	AmazonEC2	EU-DataTransfer-Regional-Bytes VPCPeering-In
3	9,527.78	AmazonEC2	EU-DataTransfer-Regional-Bytes InterZone-Out
4	6,925.35	AmazonMSK	EU-DataTransfer-Regional-Bytes Bandwidth
5	6,332.62	AWSELB	EU-DataTransfer-Regional-Bytes LoadBalancing

Peering costs money (with x-AZ/x-Region traffic)



VPC Peering Connections are free **BUT**
cross-AZ AND cross-region data transfer charges still apply!

Communication Pattern - VPC(s) to Kafka

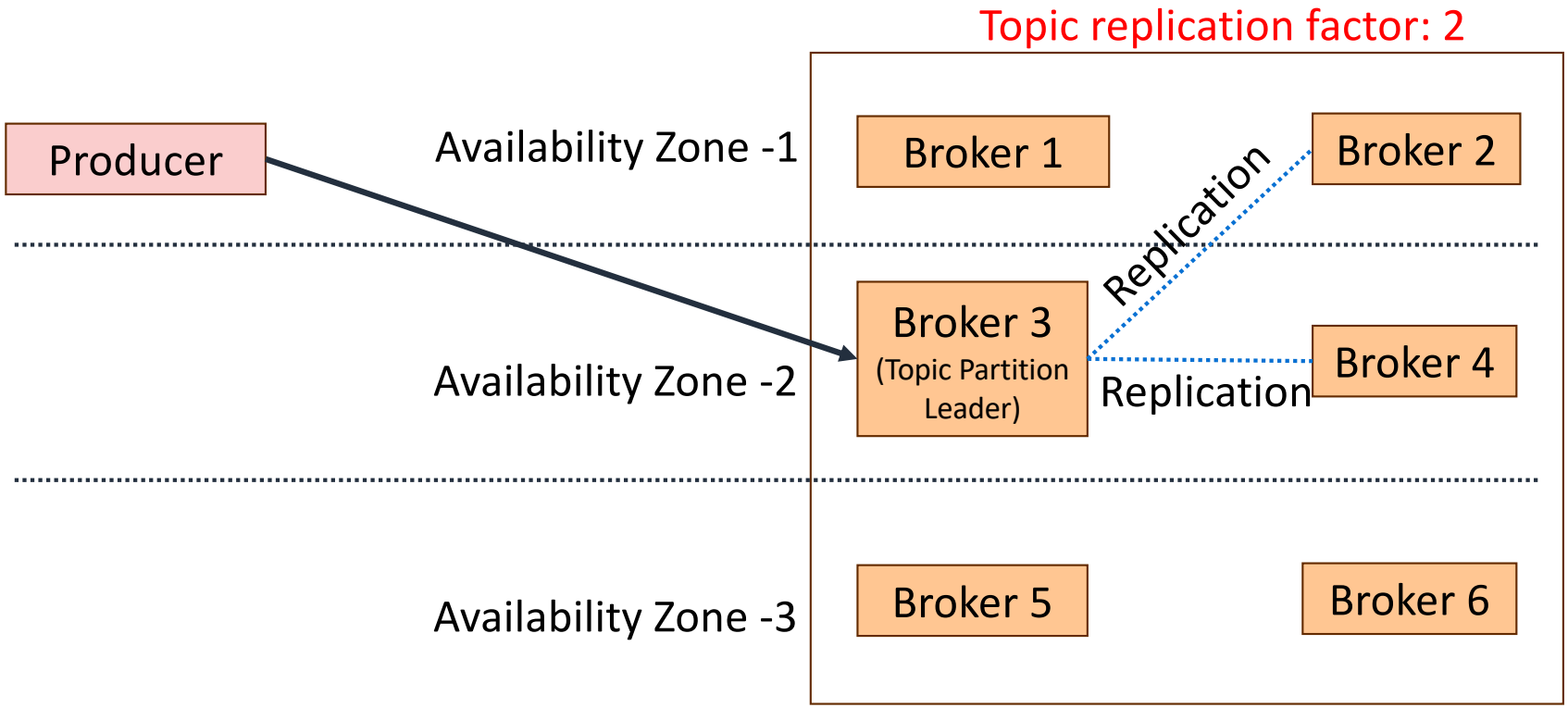


VPC Peering Limitations

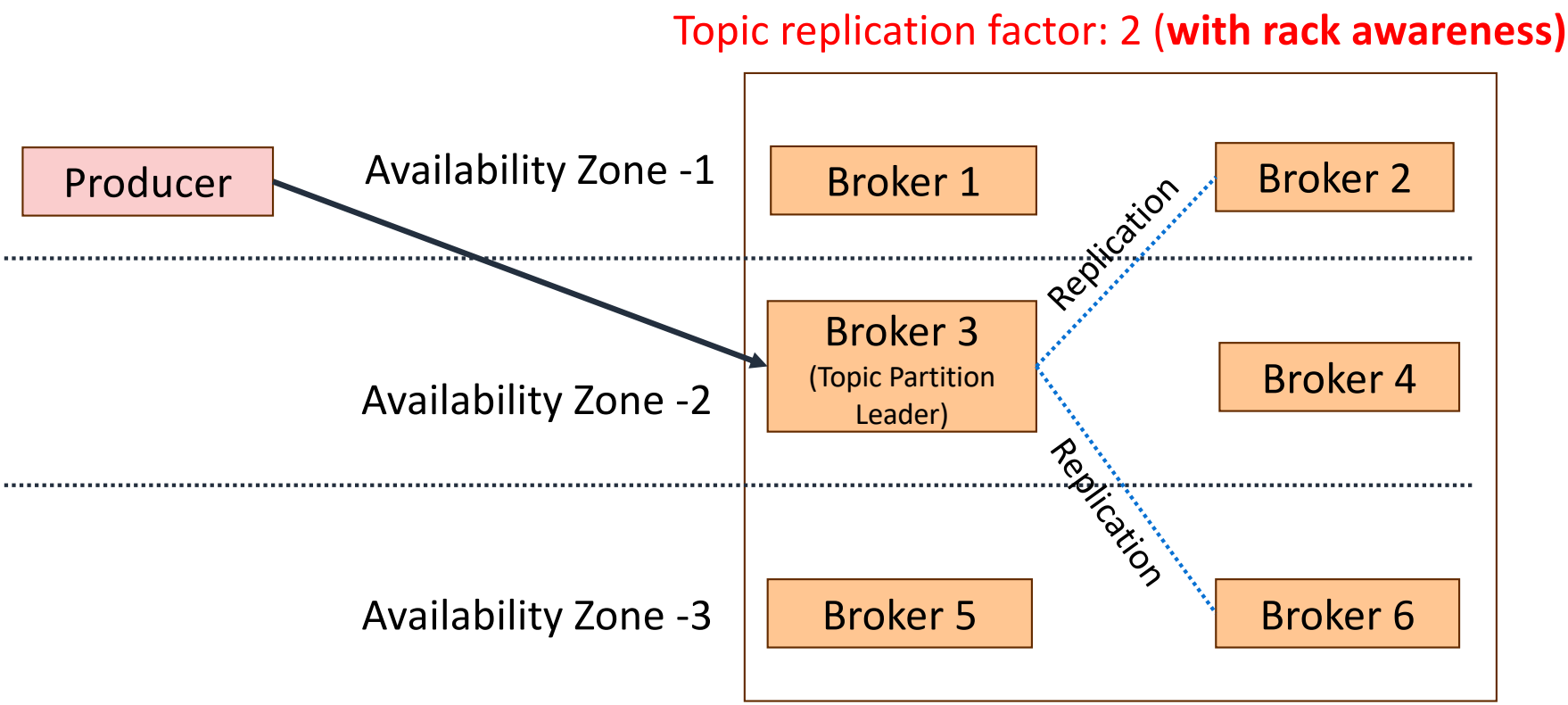
- No transitive routing
- Overlapping IP Address not supported
- Scalability and Mgmt. complexity
- Chances of NAU exhaustion
  (Many AWS users miss this!)

Overuse of VPC Peering Connections can introduce operational overheads and impact reliability!

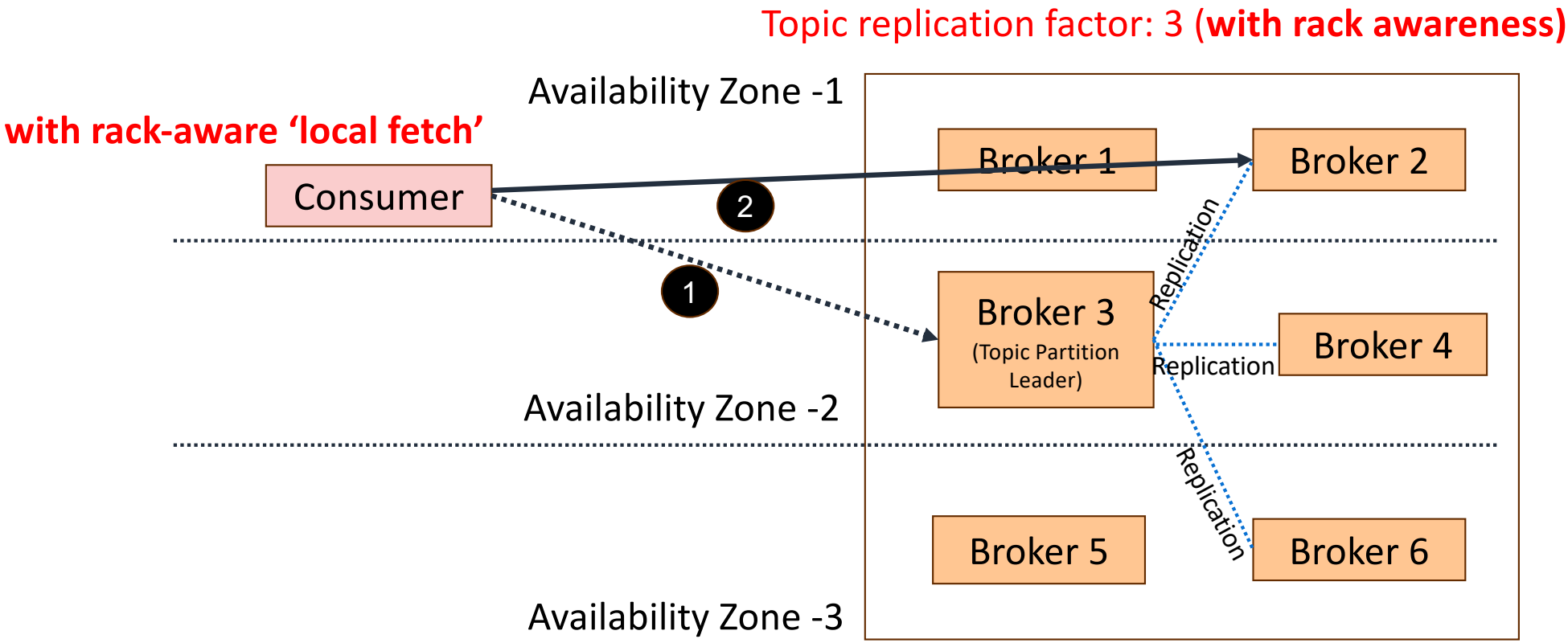
Let’s look at Kafka internals!



Kafka *topic* replication with “zone awareness”



“local fetch” to reduce cross-AZ transfers



Ensuring **“zone-awareness”** in workloads will **reduce/eliminate data transfer costs**

Note: AZ names like ***eu-central-1a*** can point to **different zone IDs** (**euc1-az1**)across different AWS accounts)

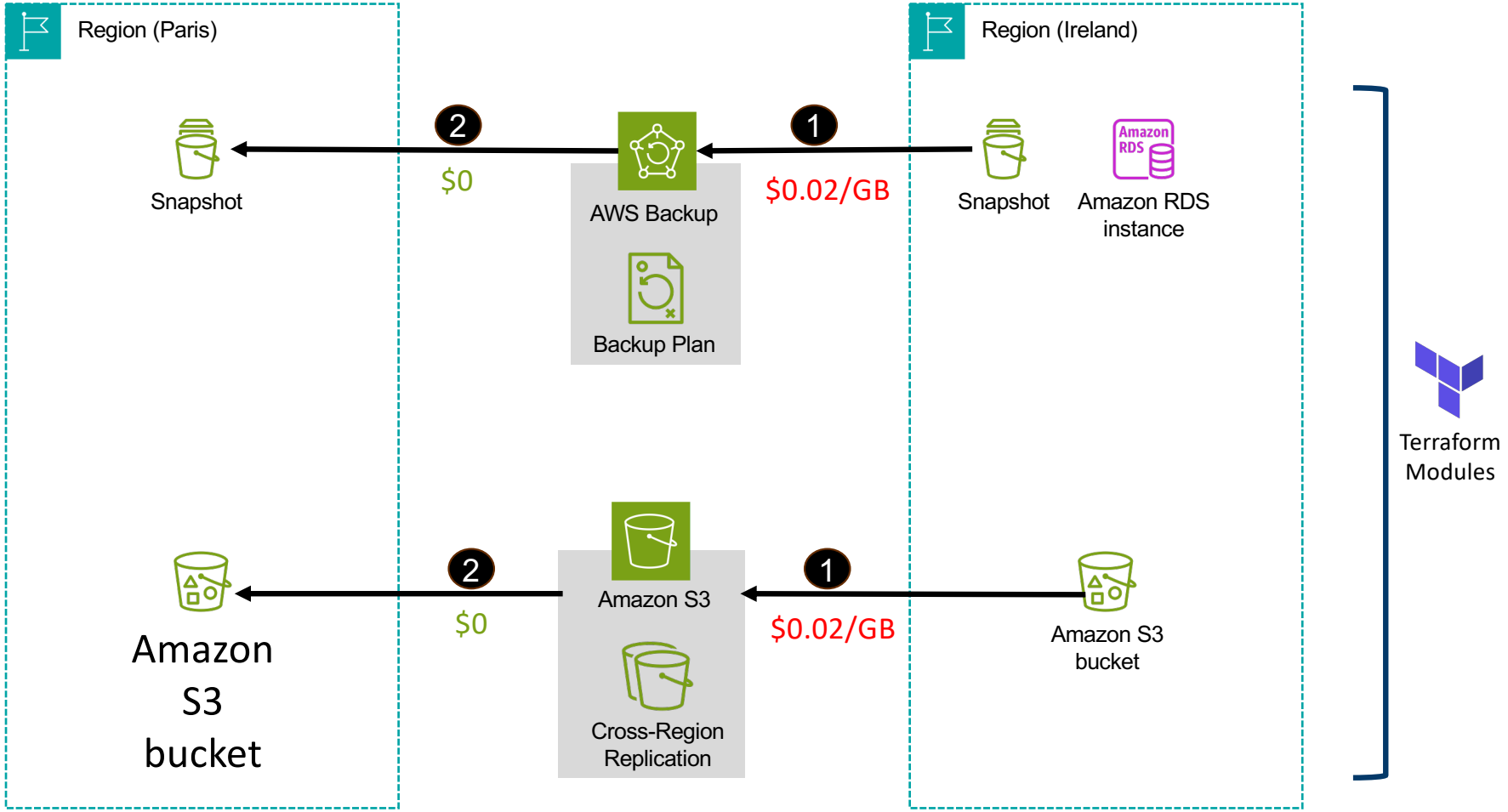
Data transfer costs
(Inter-Region)

1	IntraRegion	102,281.53
2	InterRegion Outbound	37,093.13
3	AWS Outbound	17,766.04
4	Inter Region Peering Data Transfer Outbound	3,619.92
5	CloudFront to Origin	9.79456E-5
6	IntraRegion-xAZ-In	0.0

Dissecting the "*Inter Region Outbound*" costs

	Cost	Product Code	Charge Type	Source	Destination
1	12,941.24	AmazonRDS	EU-EUW3-AWS-Out-Bytes	EU (Ireland)	EU (Paris)
2	11,875.01	AmazonS3	EU-EUC1-AWS-Out-Bytes	EU (Ireland)	EU (Frankfurt)
3	9,944.15	AmazonRDS	EU-EUW3-AWS-Out-Bytes	EU (Ireland)	EU (Paris)

Data Transfer Costs resulting from S3 replication and AWS Backup



AWS **service-initiated operations**
can also contribute to data transfer costs!

NAT Gateway costs

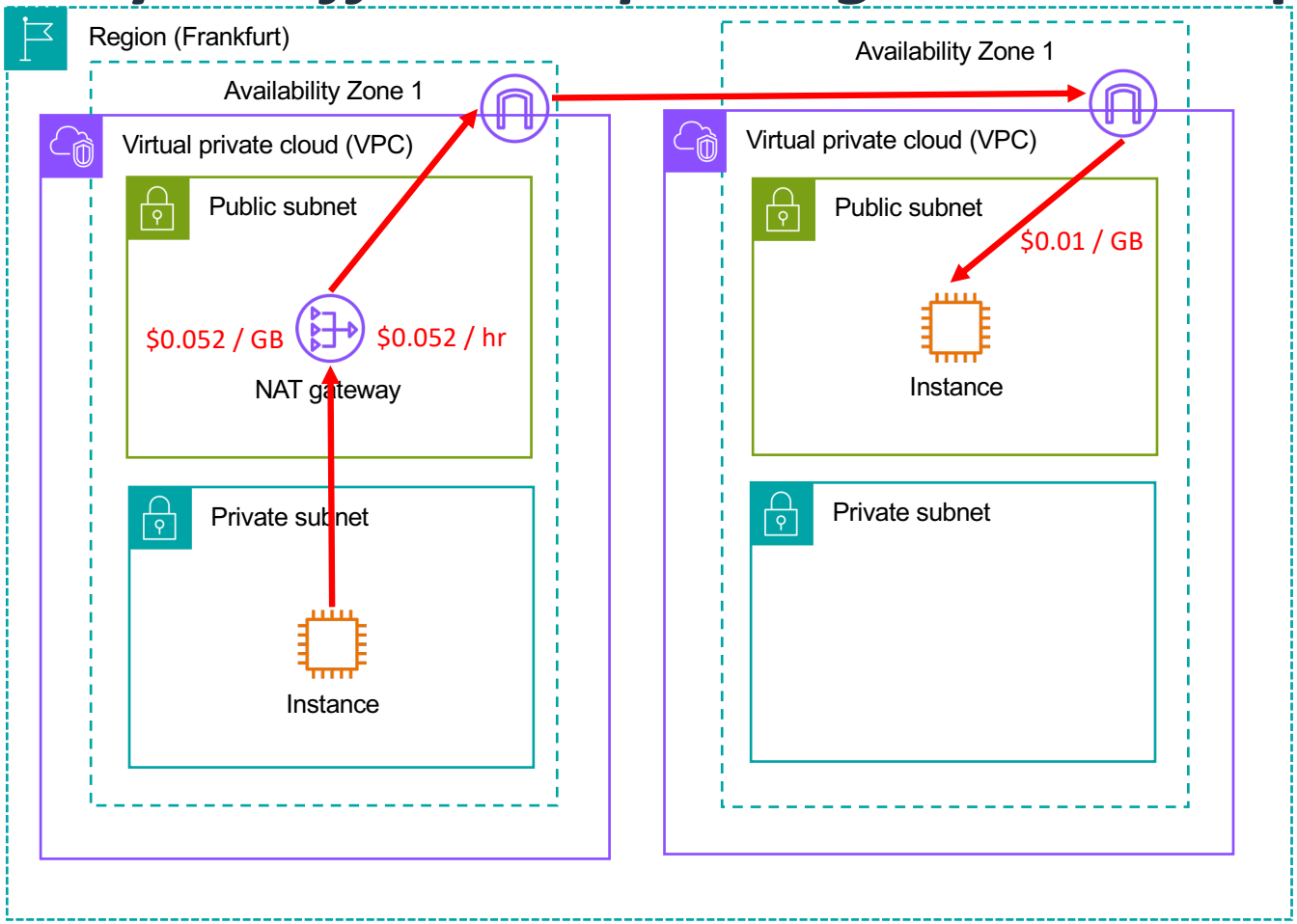
1	*NAT Gateway 50,327.68*

Dissecting "*NAT Gateway*" pricing

	Charge Type	Total Usage	Cost
1	NAT Gateway Hourly Charge	117211.0	5,623.89
2	NAT Gateway Data Transfer In	138117.30	0.00
3	NAT Gateway Data Processing Charge	806472.53	38,710.68
4	NAT Gateway Data Transfer Same Region	422680.62	4,226.80
5	NAT Gateway Data Transfer Out	72670.58	1,766.31

Traffic Hair-pinning!

Why "*Traffic Hair-pinning*" is an anti-pattern!



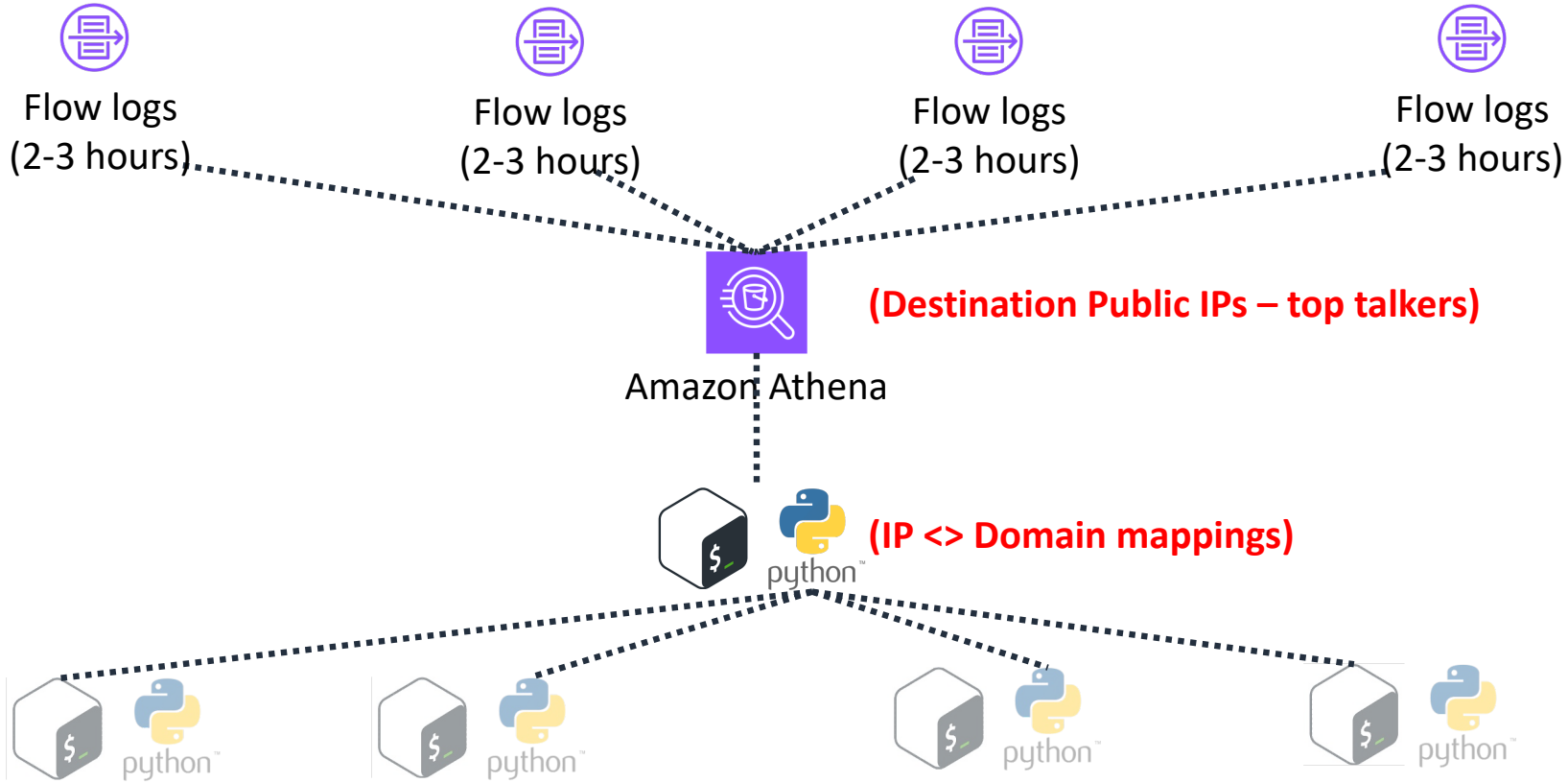
Added Latency

Data Processing Costs

Service charges for additional components

Using Public-IPs for Intra-region traffic is an anti-pattern.
Incurs an additional $0.01 - in each direction!

What destinations are NAT Gateways reaching out to?



A lot of these domains made sense

 *'104.17.3.109', '104.17.4.109'*

***.facebook.com** '157.240.221.18'*

***y.ssl-386-default.ssl.fastly.net** '199.232.25.130'*

***.blob.core.windows.net** '20.150.76.228'*

 *'3.233.145.102', '3.233.145.106', '3.233.145.111', '3.233.145.113', '3.233.145.122', '3.233.147.15', '3.233.147.19', '3.233.147.29', '3.233.147.30', '3.233.147.31', '3.233.150.112', '3.233.150.117', '3.233.150.118', '3.233.156.98', '3.233.157.164', '3.233.157.165', '3.233.157.20', '3.233.157.22', '3.233.157.34', '3.233.157.4', '3.233.157.6'*

 *'34.107.65.220'*

***.ravelin.com** '34.111.112.107'*

***ingest.sentry.io** '34.120.195.249'*

***api.stripe.com** '34.240.123.193', '34.241.202.139', '34.241.54.72', '34.241.59.225', '34.250.89.120'*

***sentry.io** '35.186.247.156'*

***ts01-b.cloudsink.net** '50.18.194.39', '54.183.140.32', '54.183.142.105', '54.241.186.124', '54.241.197.58', '54.67.119.89', '54.67.123.234', '54.67.26.184', '54.67.48.56', '54.67.54.116', '54.67.68.88', '54.67.92.206'*

 *'52.95.118.165', '54.239.32.126', '54.239.37.73', '67.220.224.163', '67.220.226.247', '67.220.228.229'*

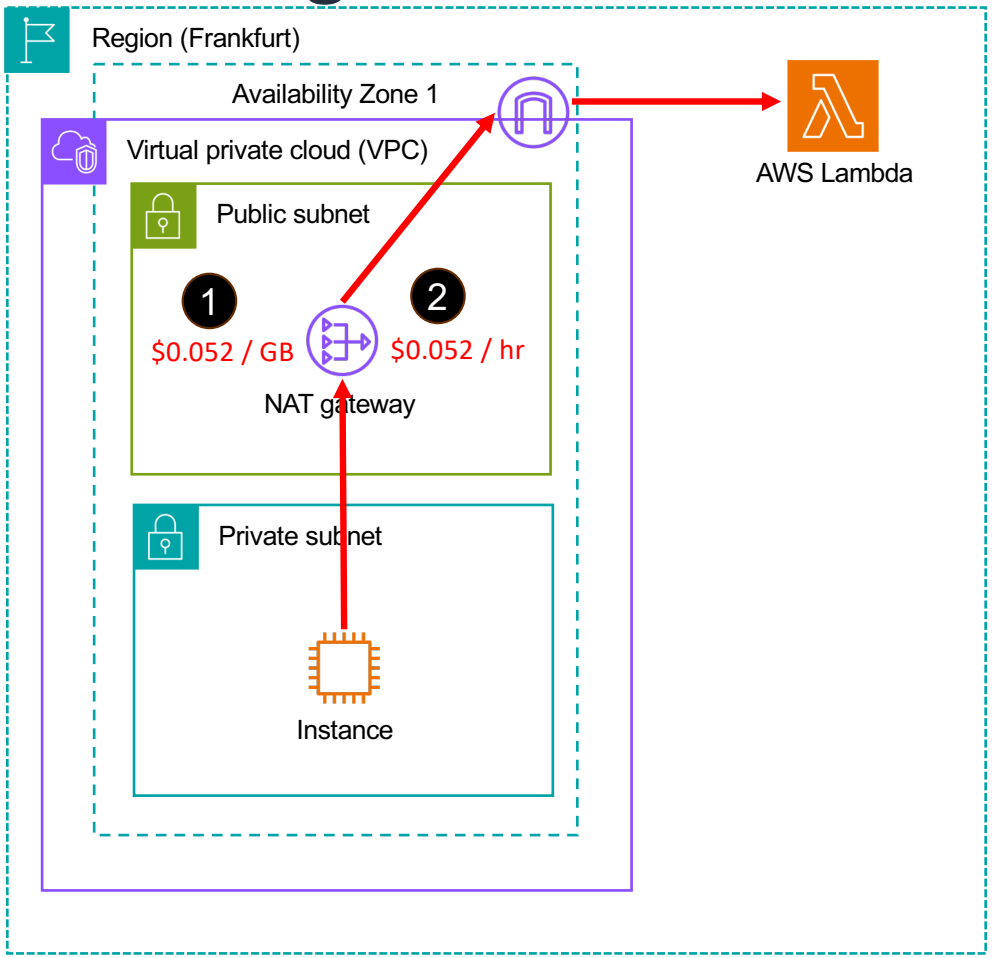
But why Lambda!

lambda.eu-west-1.amazonaws.com *'54.247.236.111', '54.247.236.123', '54.247.236.126', '54.247.236.128', '54.247.236.145', '54.247.236.153', '54.247.236.168', '54.247.236.172', '54.247.236.176', '54.247.236.221', '54.247.236.232', '54.247.236.243', '54.247.236.245', '54.247.236.3', '54.247.236.40', '54.247.236.41', '54.247.236.45', '54.247.236.93', '54.247.236.94', '54.247.237.104', '54.247.237.12', '54.247.237.120', '54.247.237.127', '54.247.237.13', '54.247.237.157', '54.247.237.160', '54.247.237.164', '54.247.237.169', '54.247.237.177', '54.247.237.190', '54.247.237.20', '54.247.237.203', '54.247.237.212', '54.247.237.221', '54.247.237.228', '54.247.237.252', '54.247.237.39', '54.247.237.40', '54.247.237.41', '54.247.237.59', '54.247.237.73', '54.247.237.85', '54.247.237.98', '54.247.238.10', '54.247.238.12', '54.247.238.158', '54.247.238.16', '54.247.238.161', '54.247.238.189', '54.247.238.212', '54.247.238.223', '54.247.238.235', '54.247.238.24', '54.247.238.28', '54.247.238.35', '54.247.238.42', '54.247.238.49', '54.247.238.6', '54.247.238.66', '54.247.238.69', '54.247.238.8', '54.247.239.1', '54.247.239.101', '54.247.239.119', '54.247.239.172', '54.247.239.174', '54.247.239.201', '54.247.239.220', '54.247.239.236', '54.247.239.24', '54.247.239.242', '54.247.239.244', '54.247.239.250', '54.247.239.255', '54.247.239.3', '54.247.239.44', '54.247.239.56', '54.247.239.75', '54.247.239.89', '63.32.72.113', '63.32.72.116', '63.32.72.117', '63.32.72.120', '63.32.72.128', '63.32.72.134', '63.32.72.137', '63.32.72.142', '63.32.72.156', '63.32.72.167', '63.32.72.170', '63.32.72.175', '63.32.72.179', '63.32.72.183', '63.32.72.207', '63.32.72.212', '63.32.72.213', '63.32.72.223', '63.32.72.225', '63.32.72.233', '63.32.72.237', '63.32.72.238', '63.32.72.247', '63.32.72.255', '63.32.72.70', '63.32.73.102', '63.32.73.116', '63.32.73.119', '63.32.73.12', '63.32.73.124', '63.32.73.130', '63.32.73.138', '63.32.73.143', '63.32.73.146', '63.32.73.148', '63.32.73.154', '63.32.73.158', '63.32.73.161', '63.32.73.165', '63.32.73.168', '63.32.73.17', '63.32.73.179', '63.32.73.182', '63.32.73.187', '63.32.73.188',*

~1/3rd of NAT G/w traffic were Lambda invocations!

<REDACTED>	0.34 GB	Investigate if non-public connectivity options are available
*.facebook.com	2.43 GB	
y.ssl-386-default.ssl.fastly.net	1.46 GB	
*.blob.core.windows.net	0.6 GB	
<REDACTED>	2.12 GB	
<REDACTED>	4.24 GB	<same_as_above>
<REDACTED>	4.63 GB	<same_as_above>
ingest.sentry.io	1.65 GB	
api.stripe.com	0.53 GB	<same_as_above>
sentry.io	2.51 GB	<same_as_above>
ts01-b.cloudsink.net	2.19 GB	<same_as_above>
<REDACTED>	1.02 GB	
lambda.eu-west-1.amazonaws.com	218.75 GB	32.68% of NAT Gateway Traffic volume (~from flow logs' capture of 2-3 hours).

Accessing Lambda service endpoints via NAT Gateway

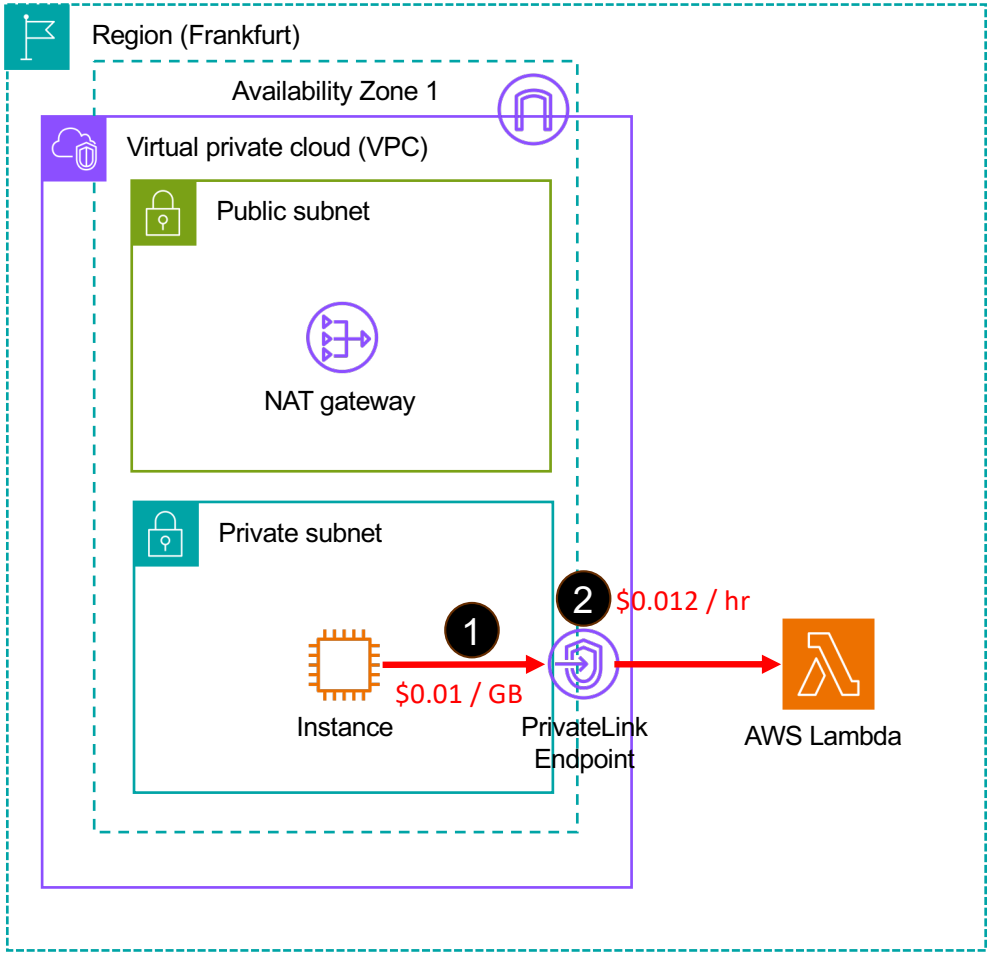


1. 156,960 GB data transfer
from EC2-1 >> Lambda
$0.052/GB **NAT Processing**
= $8161

2. 730 hours
$0.052/hr **NAT Service Usage**
= $38

Total = $8199

Accessing Lambda service endpoints via PrivateLink



1. 156,960 GB data transfer
   from EC2-1 >> Lambda
   $0.01/GB **PrivateLink Processing**
   = $1569

2. 730 hours
   $0.012/hr **NAT Service Usage**
   = $8

Total = $1577 (81% savings)

Use PrivateLink for accessing AWS Service Endpoints!

Key takeaways

Learnings

1. The **AWS region(s)** you choose to host your workloads can **influence your data transfer costs**

2. **VPC Peering Connections** are free BUT **cross-AZ AND cross-region data transfer charges** still apply

3. **Avoid overuse of VPC Peering Connections** can introduce operational overheads and impact reliability

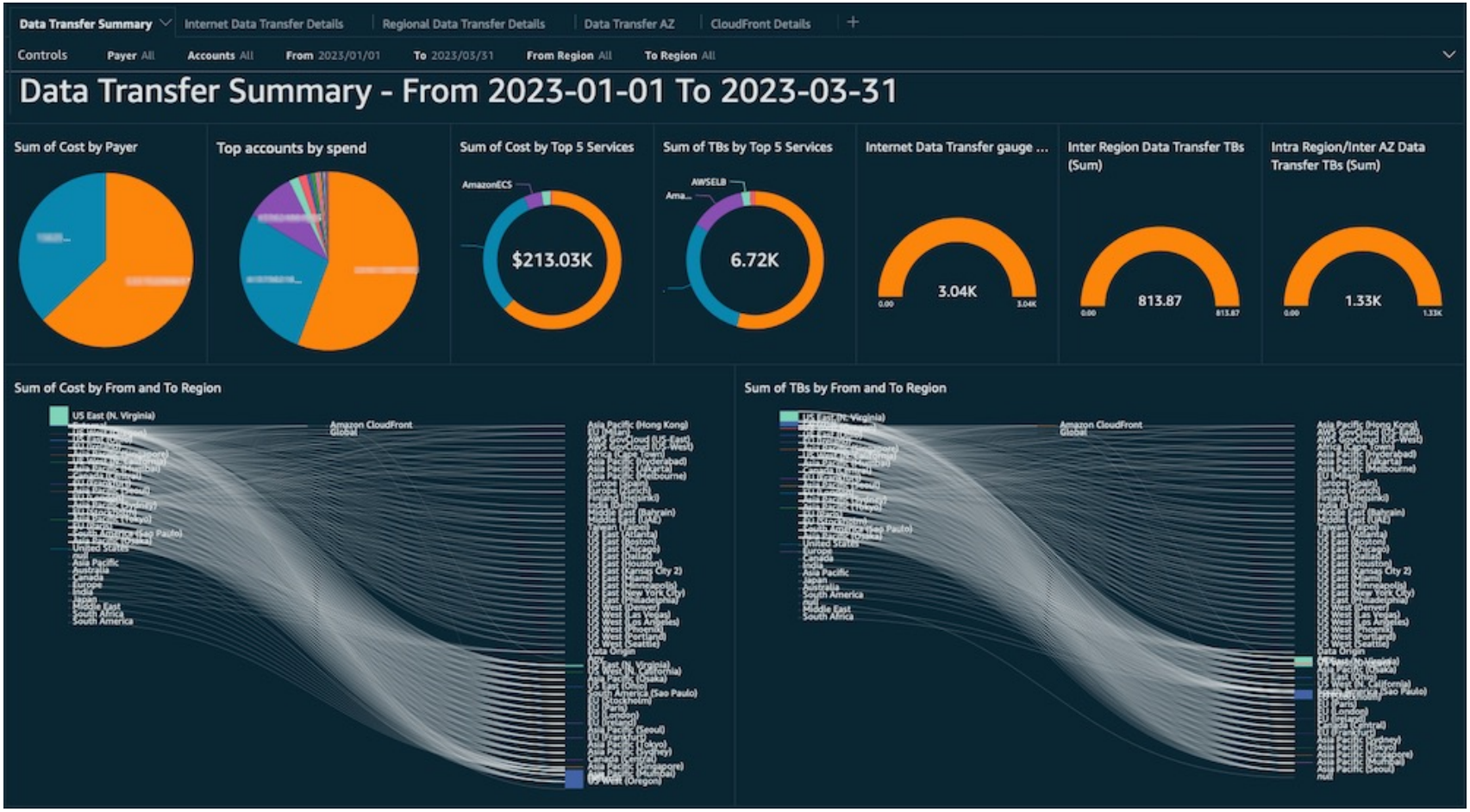
4. Ensuring **“zone-awareness”** in workloads can **reduce data transfer costs**

5. AWS **service-initiated operations** can also contribute to **data transfer costs**

6. **Public-IPs for Intra-region** traffic is an **anti-pattern**. Incurs an additional $0.01 (both ways)

7. Always prefer using **PrivateLink for accessing AWS Service Endpoints**

AWS Data Transfer Dashboard



Thank you!

Questions are welcome ☺

Akshay Kapoor

Professional PainKiller | Author of AWS DevOps Simplified | DevOps | AWS | Cloud