
AWS
re:Invent

NET404-R

How to test network performance on AWS

Theodoros Stamatopoulos
Sr. Cloud Support Engineer
Amazon Web Services

AWS re:Invent

aws

Agenda

Constants and variables in AWS networking world

Test bandwidth performance

Constants and variables in networking world

Test PPS performance

Preparation

- Download the document on link: **https://bit.ly/2qXbH2b**
- Main Region – Please use either Ireland, London, or Frankfurt as your main Region
- Remote Region – Please use either Oregon, Singapore, or Bahrain as your remote

Constants and variables in AWS networking

AWS re:Invent

aws

Constant

Instances within a cluster placement group can use up to 10 Gbps for single-flow traffic. Instances that are not within a cluster placement group can use up to 5 Gbps for single-flow traffic.

Variable

Each instance type and size has its own network performance characteristics

Constant

Enhanced networking provides better packet-per-second rate

Constant

Instance documented network performance of 10 Gbps

Variable

Instance documented network performance of “up to 10 Gbps”

Network performance burst

Instances are allowed to achieve more bandwidth for a limited amount of time; after that, performance is limited to a baseline that depends on instance size and type

Variable

Each instance generation has its own performance characteristics

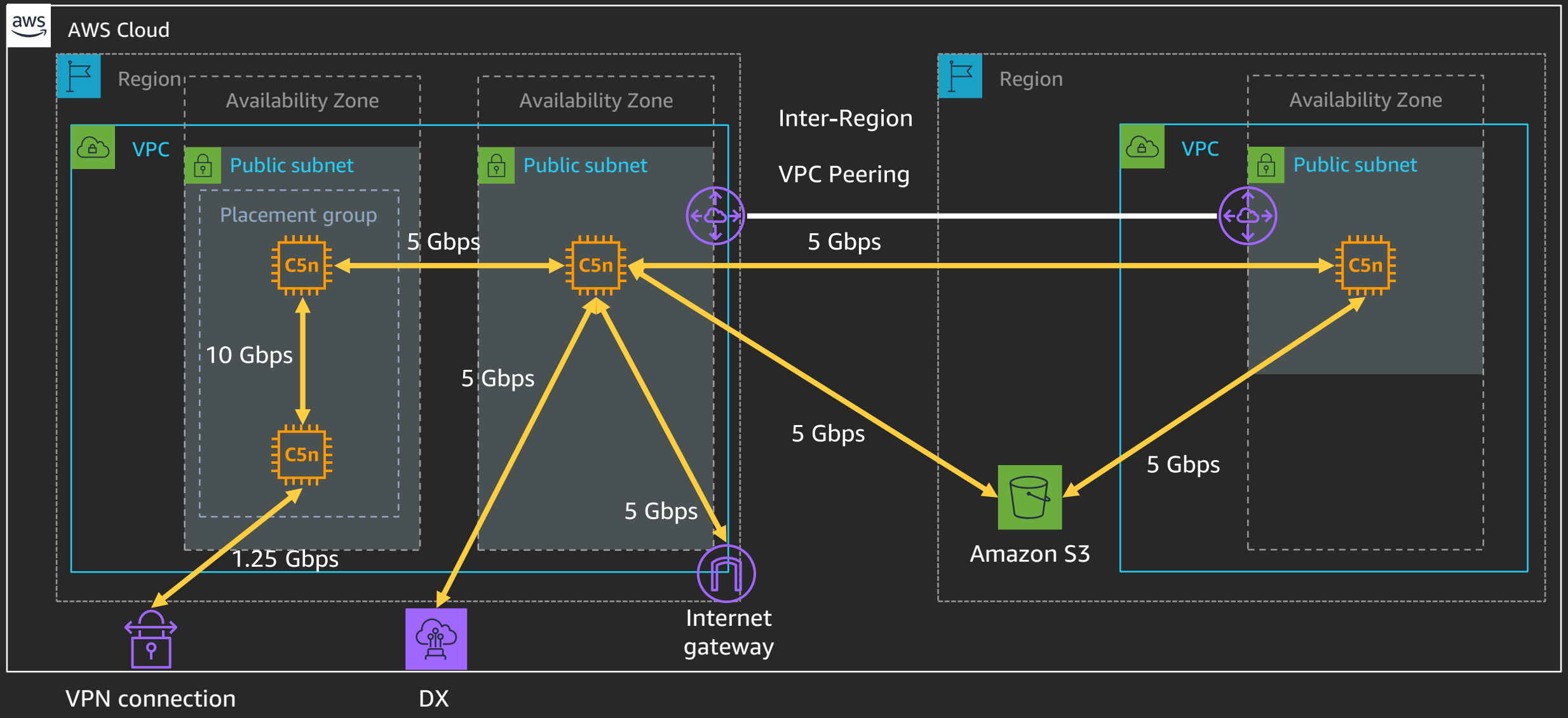
Constant

AWS Site-to-Site VPN offers 1.25 Gbps

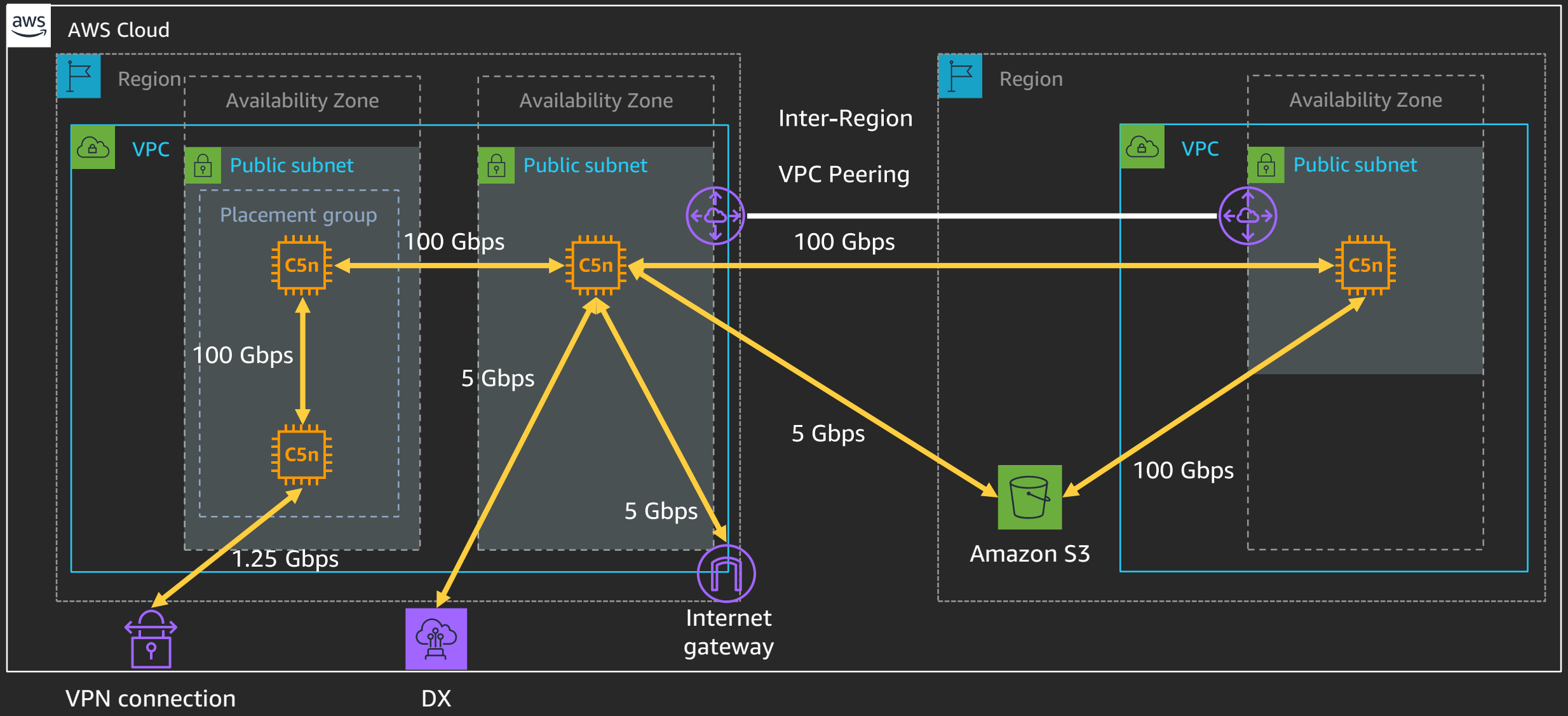
Constant

AWS Direct Connect port speed

Single-flow maximum bandwidth on C5n.18xlarge



Multiple flows maximum bandwidth on C5n.18xlarge



Tooling

Tooling – iPerf 3 overview

- Cross-platform
- TCP and UDP traffic generator
- Measures bandwidth, loss, congestion window size
- Caveats
  - Windows version doesn’t automatically scale the congestion window; you have to use the parameter “-w” to specify the maximum window size

Tooling – iPerf 3 interesting options

- General options
  - -p: Set port, default 5201
  - -J: JSON output, useful for automation
- Server-only options
  - -s: Run iPerf in server mode

Tooling – iPerf 3 interesting options

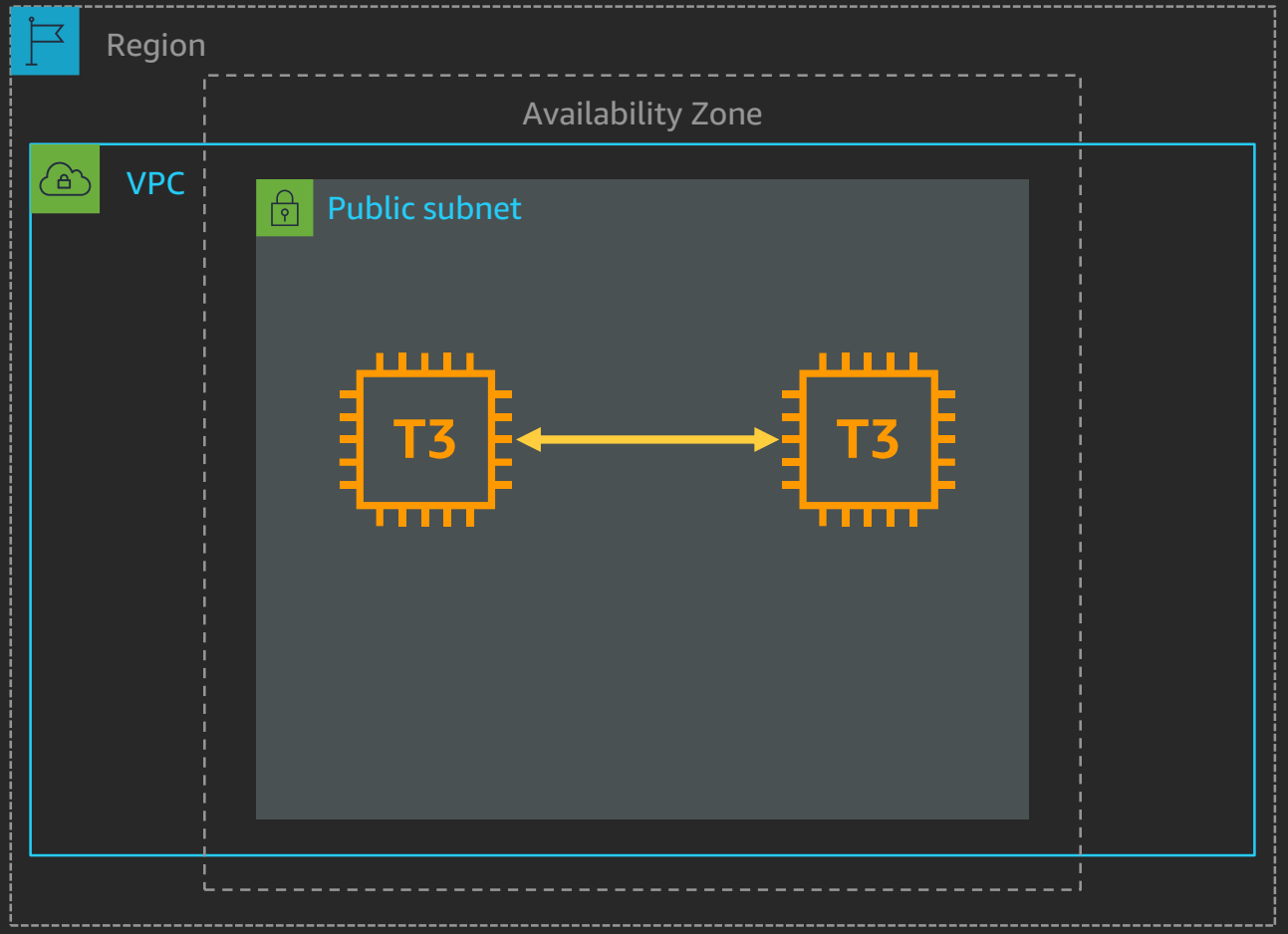
- Client-only options
  - -c: Run iPerf in client mode
  - -u: Generate UDP traffic
  - -b: Target bandwidth generation for UDP
  - -t: Time in seconds to run
  - -P: The number of simultaneous connections to make to the server
  - -R: Run in reverse mode (server sends, client receives)
  - -l: Set packet length

Using iPerf in AWS

Testbed – Things to consider before starting

- Instance generation, type, and size
- New instance
- Operating system
- Latest AWS-provided AMI fully updated
- Repeat

First scenario – Single TCP flow within Region



First scenario – Single TCP flow within Region

iPerf server

```
$ iperf3 -s
```

iPerf client

```
$ iperf3 -c <iPerf server IP> -t 120
```

First scenario – Single TCP flow within Region

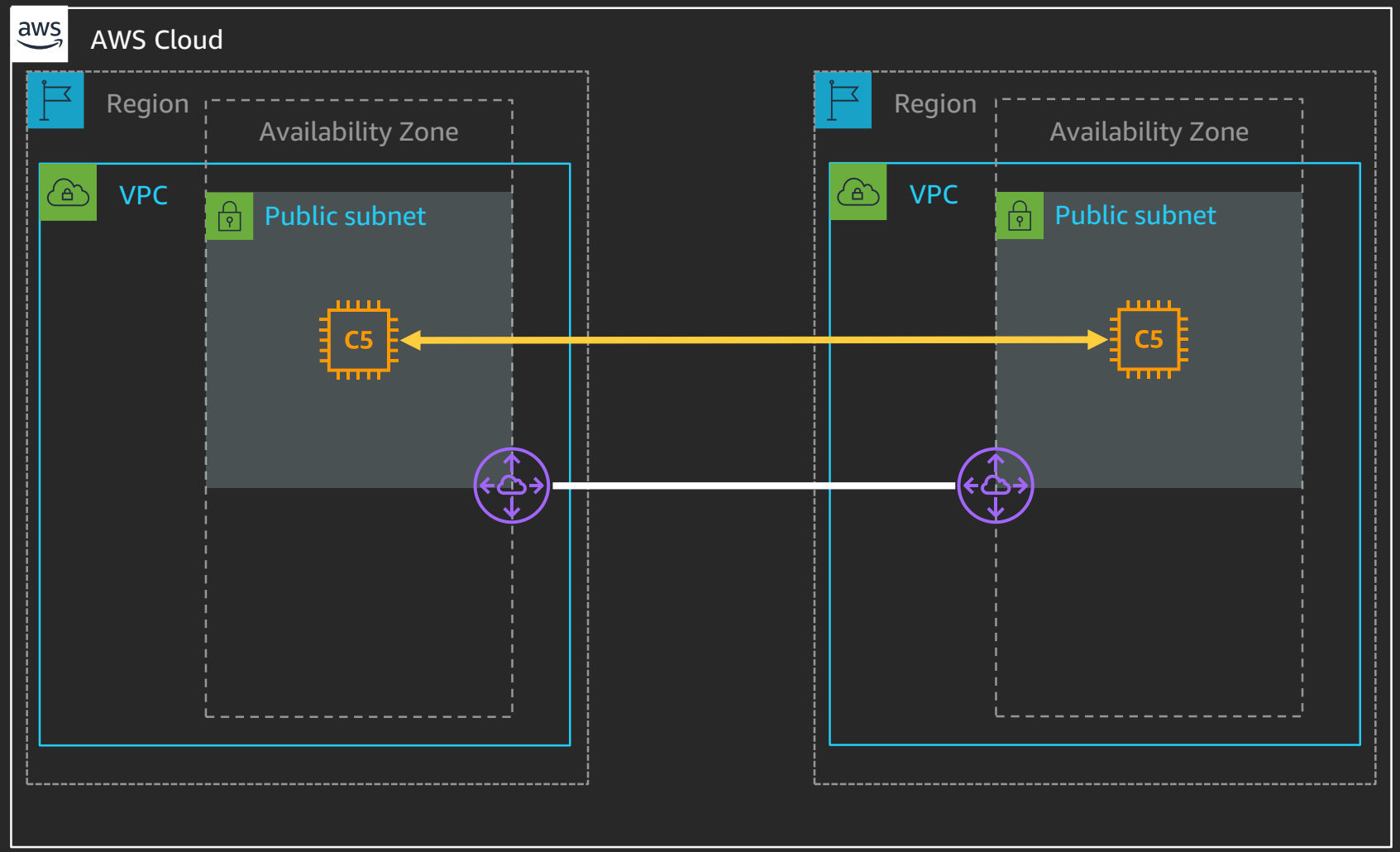
iPerf server

```
$ iperf3 -s
```

iPerf client

```
$ iperf3 -c <iPerf server IP> -t 300
```

Second scenario – Single TCP flow inter-Region



Second scenario – Single TCP flow inter-Region

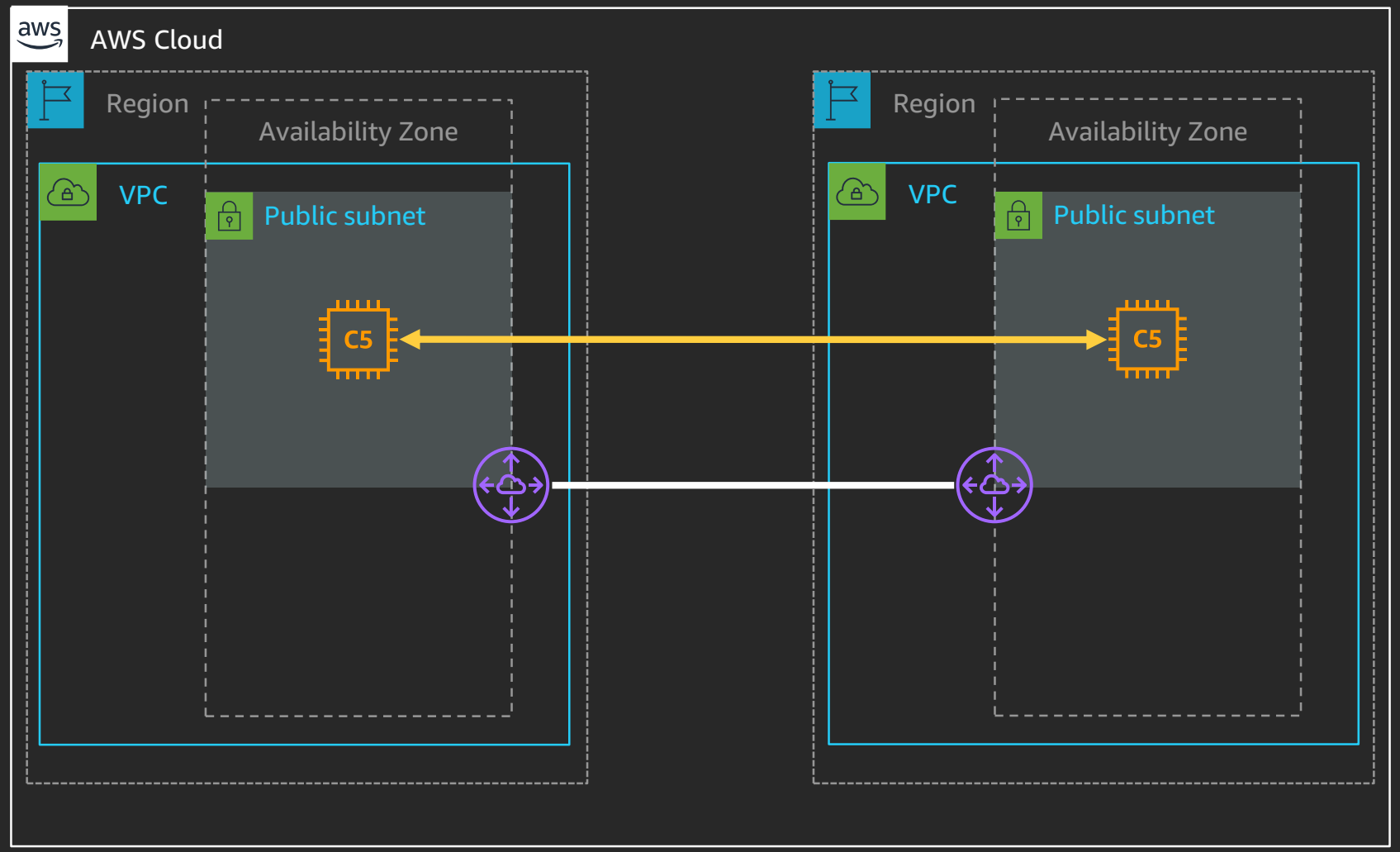
iPerf server

```
$ iperf3 -s
```

iPerf client

```
$ iperf3 -c <iPerf server IP> -t 60
```

Third scenario – Multiple TCP flows inter-Region



Third scenario – Multiple TCP flows inter-Region

iPerf server

```
$ iperf3 -s
```

iPerf client

```
$ iperf3 -c <iPerf server IP> -t 60 -P 10
```

Linux TCP stack tuning for high-latency connections

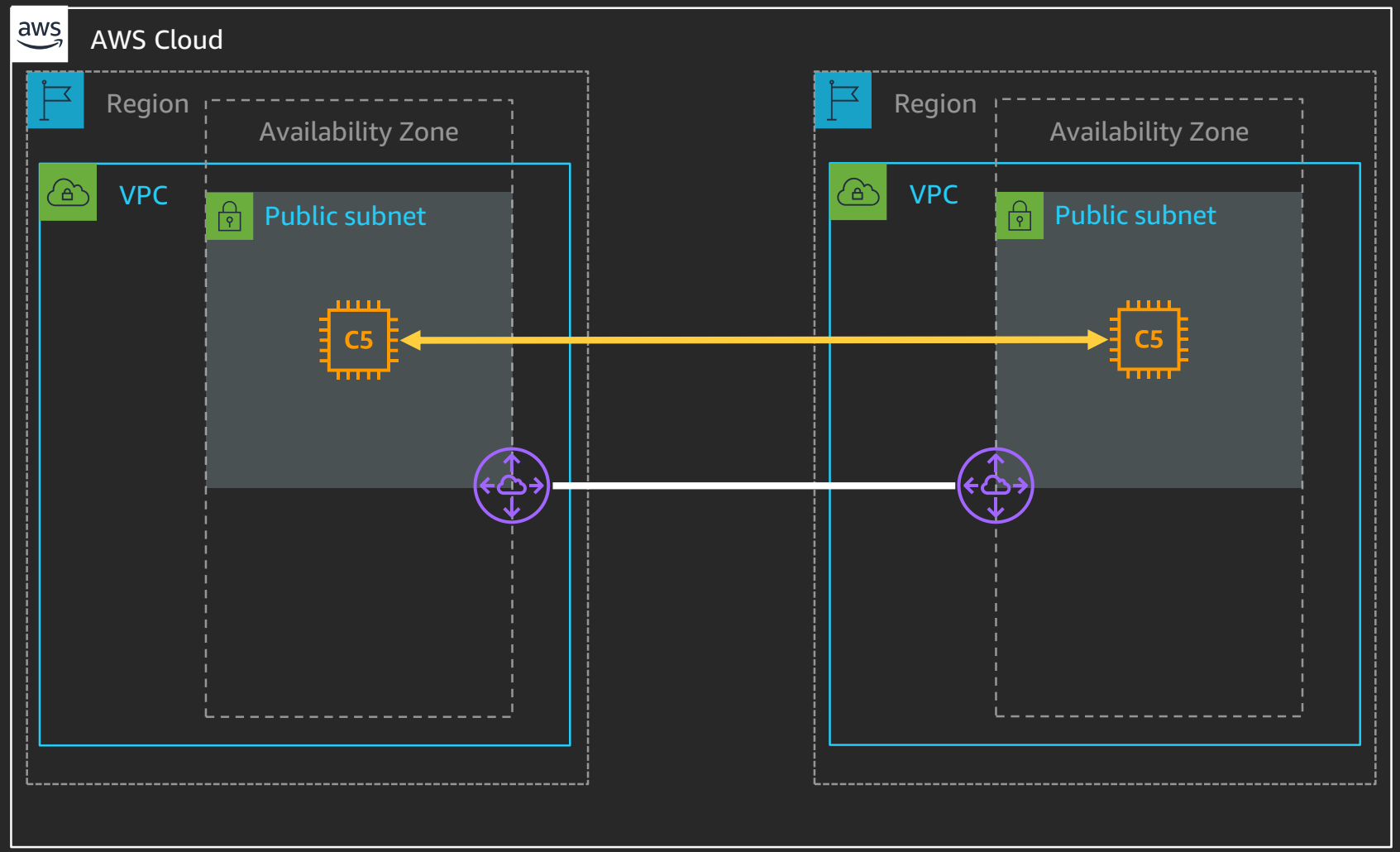
TCP buffer:

```
sudo sysctl -w net.ipv4.tcp_rmem=“4096 87380 67108864”
sudo sysctl -w net.ipv4.tcp_wmem=“4096 87380 67108864”
```

Congestion algorithm:

```
sudo sysctl -w net.ipv4.tcp_congestion_control=bbr
```

Fourth scenario – Single TCP flow inter-Region



Fourth scenario – Single TCP flow inter-Region (again)

iPerf server

```
$ iperf3 -s
```

iPerf client

```
$ iperf3 -c <iPerf server IP> -t 60
```

Congestion algorithms

Different algorithms for different use cases

Feedback mechanisms

Examples:

- Tahoe/Reno
  - One of the first congestion control mechanisms
  - Tahoe – If three duplicate acks received, sets MSS 1, restarts slow start
  - Reno – If three duplicate acks received, it sets the window to half and starts the slow start from there
  - Loss feedback

Congestion algorithms (cont.)

- Cubic
  - Linux default
  - Coming to Microsoft Windows (experimental)
  - Loss feedback
- BBR
  - Included in Linux starting from kernel 4.9
  - Delay feedback

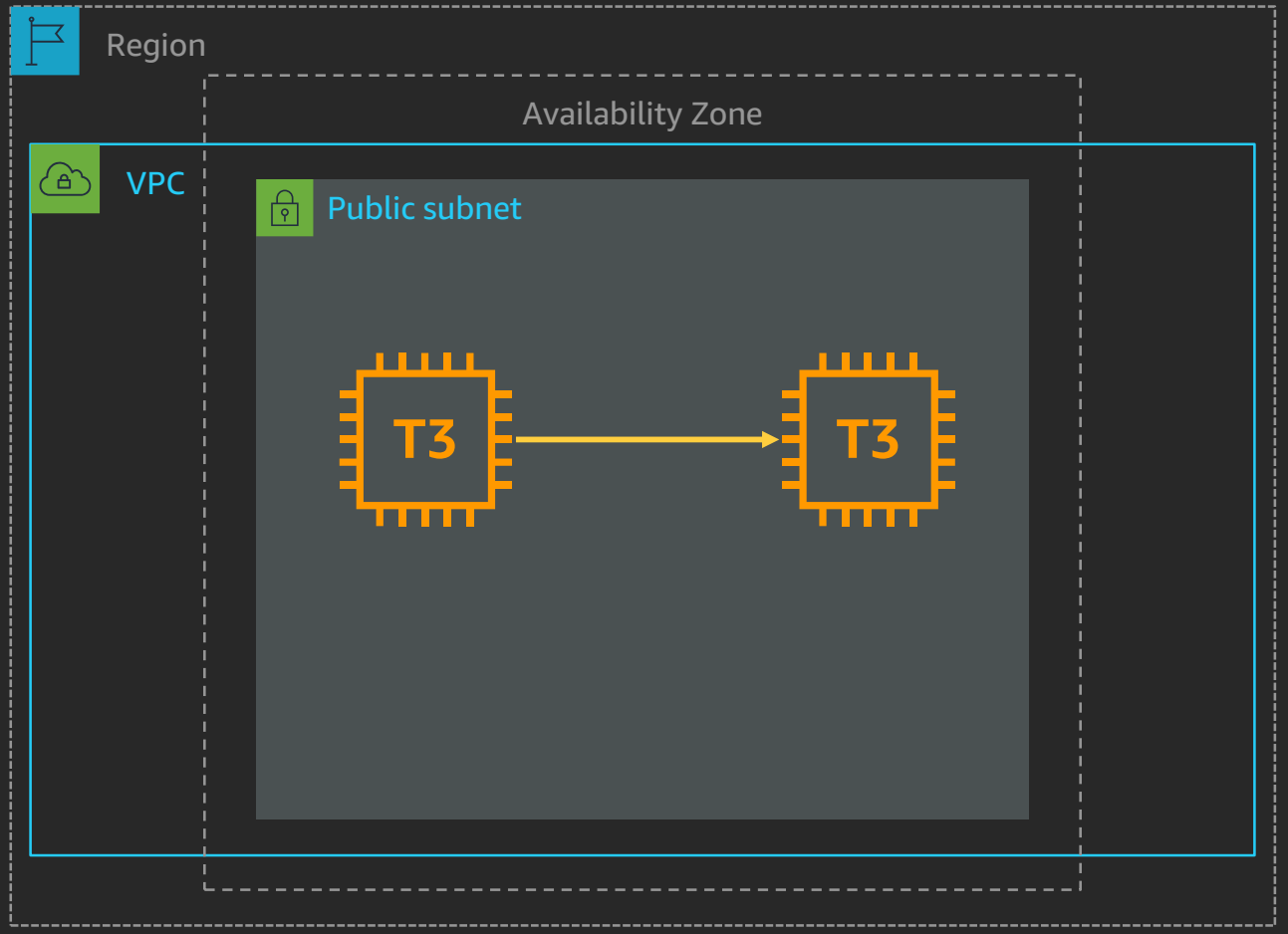
Testing packet per second

How to test packet per second

UDP doesn’t have any mechanism for flow control. It just sends traffic.

By monitoring the receiver, you can find the limit.

Fifth scenario – UDP flow, same Region



Fifth scenario – UDP flow, same Region

iPerf server :

$ iperf3 –s

iPerf client :

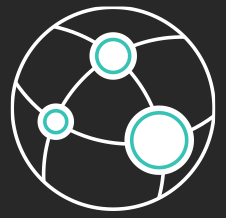
$ iperf3 -c *<iPerf server IP>* -u -b 10G –t 60 -l 64

Second terminal on the server:

$ iptraf-ng -d eth0

Learn networking with AWS Training and Certification

Resources created by the experts at AWS to help you build and validate networking skills



Free digital courses cover topics related to networking and content delivery, including Introduction to Amazon CloudFront and Introduction to Amazon VPC

Validate expertise with the
AWS Certified Advanced Networking - Specialty exam

Visit aws.amazon.com/training/paths-specialty

aws training and certification

Thank you!

Theodoros Stamatopoulos

stamatot@amazon.com

AWS re:Invent

aws

Please complete the session survey in the mobile app.