

Essential AWS for PHP Applications

Longhorn PHP 2018

Michael Moussa

@michaelmoussa

<https://joind.in/talk/97b36>



Who am I?

Who am I?

- Product Architect at Rackspace



Who am I?

- Product Architect at Rackspace
- 18 year PHP developer & DevOps engineer



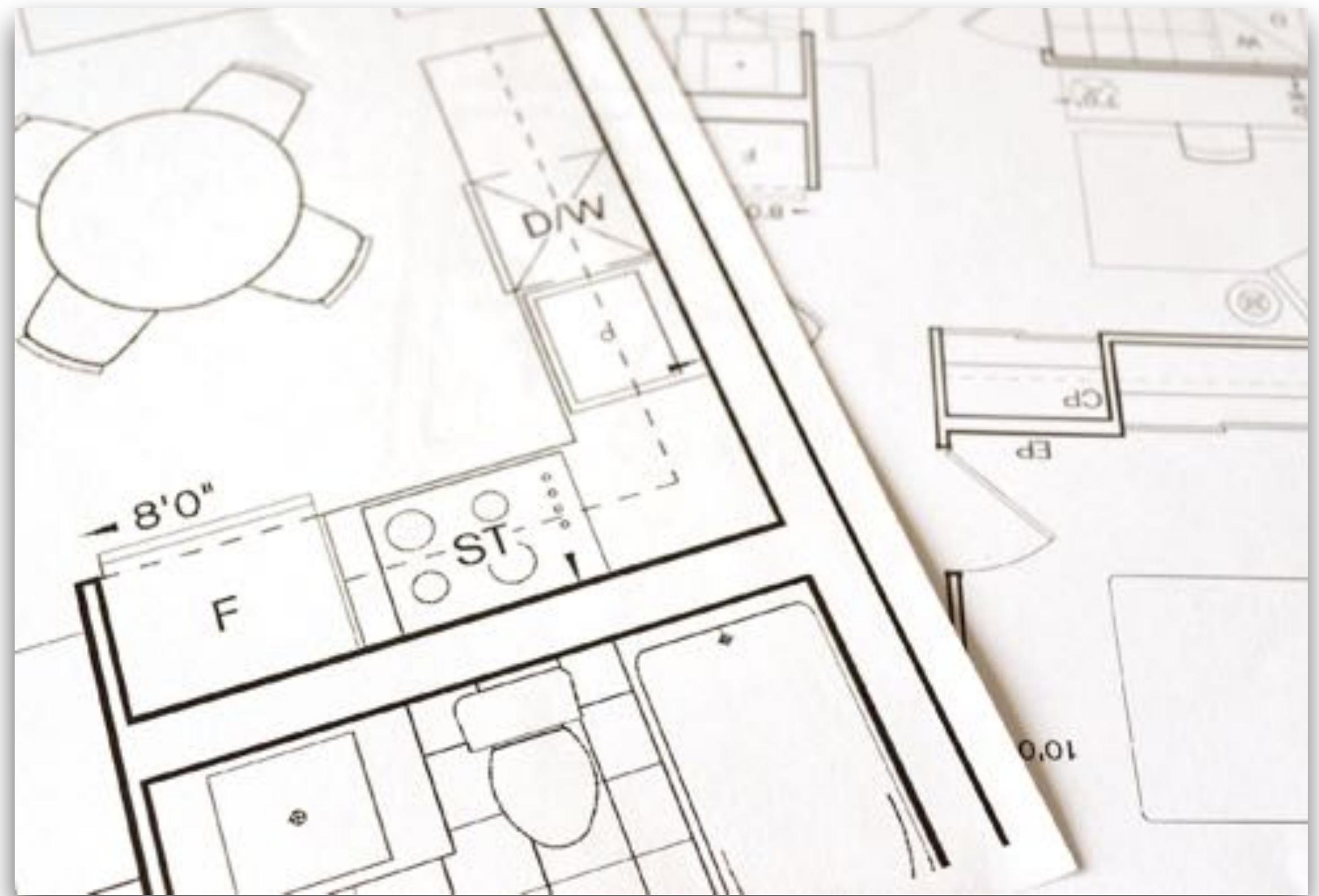
Who am I?

- Product Architect at Rackspace
- 18 year PHP developer & DevOps engineer
- AWS Certified Solutions Architect



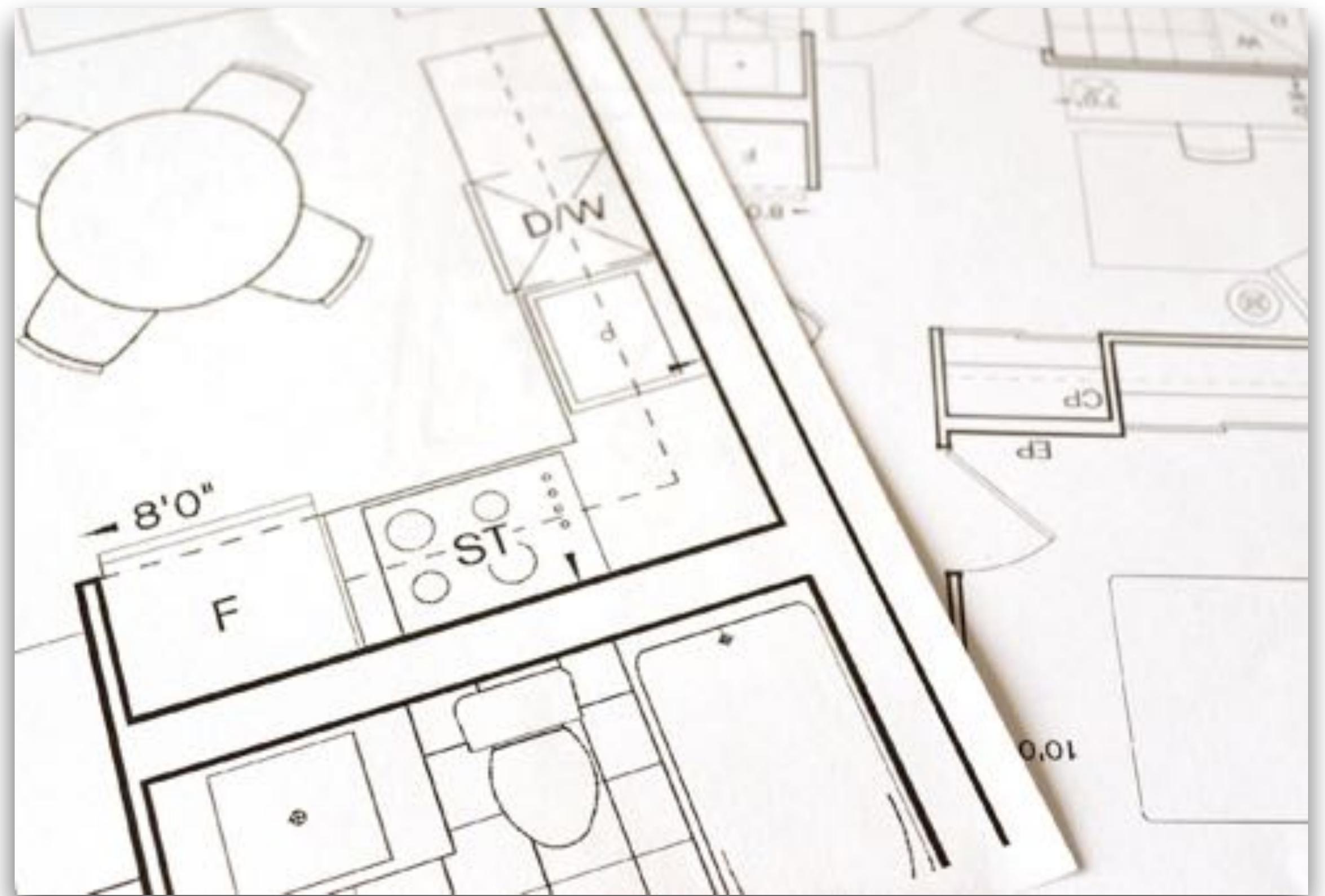
Sample environment requirements

- Run PHP applications



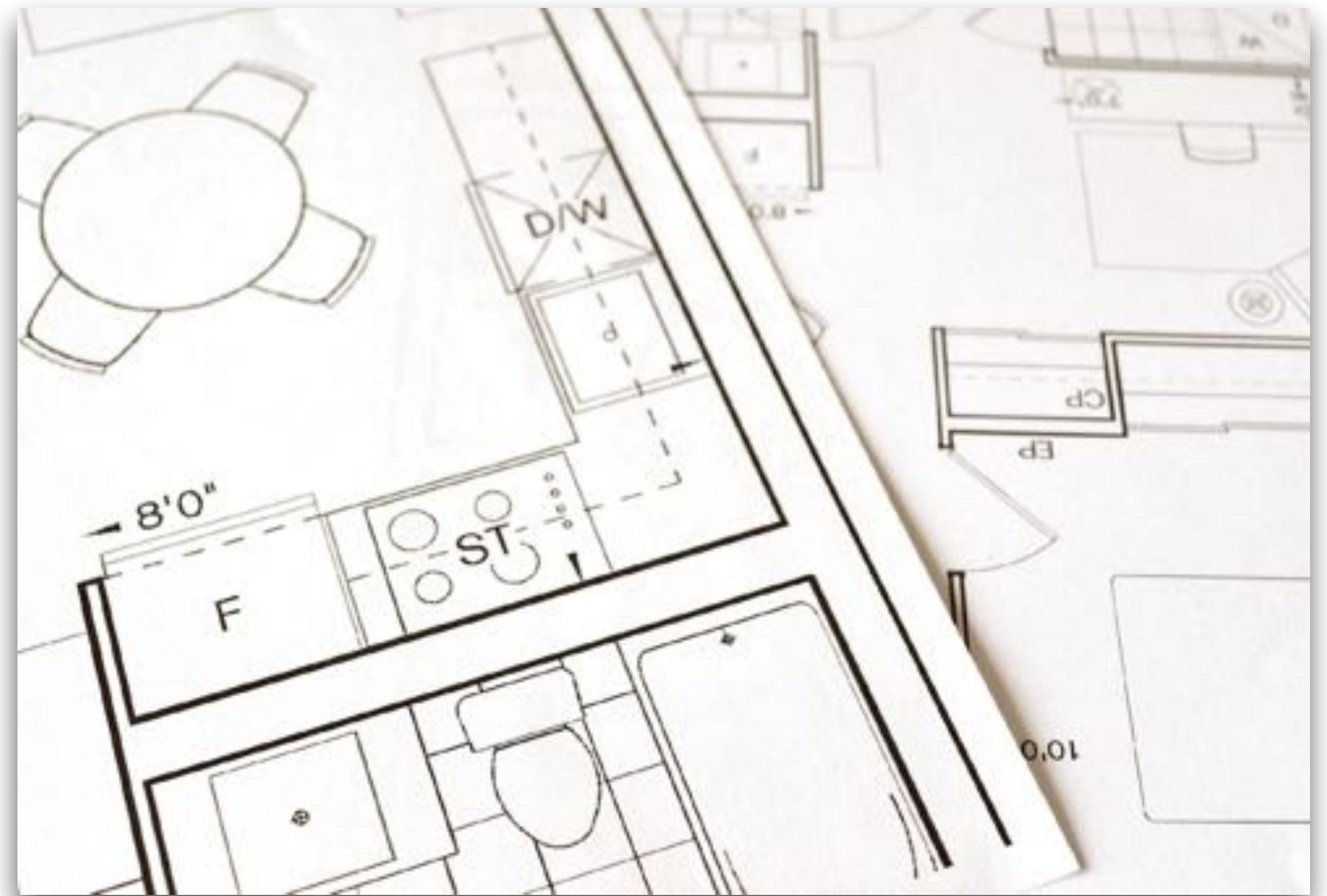
Sample environment requirements

- Run PHP applications
- Store information in a DB



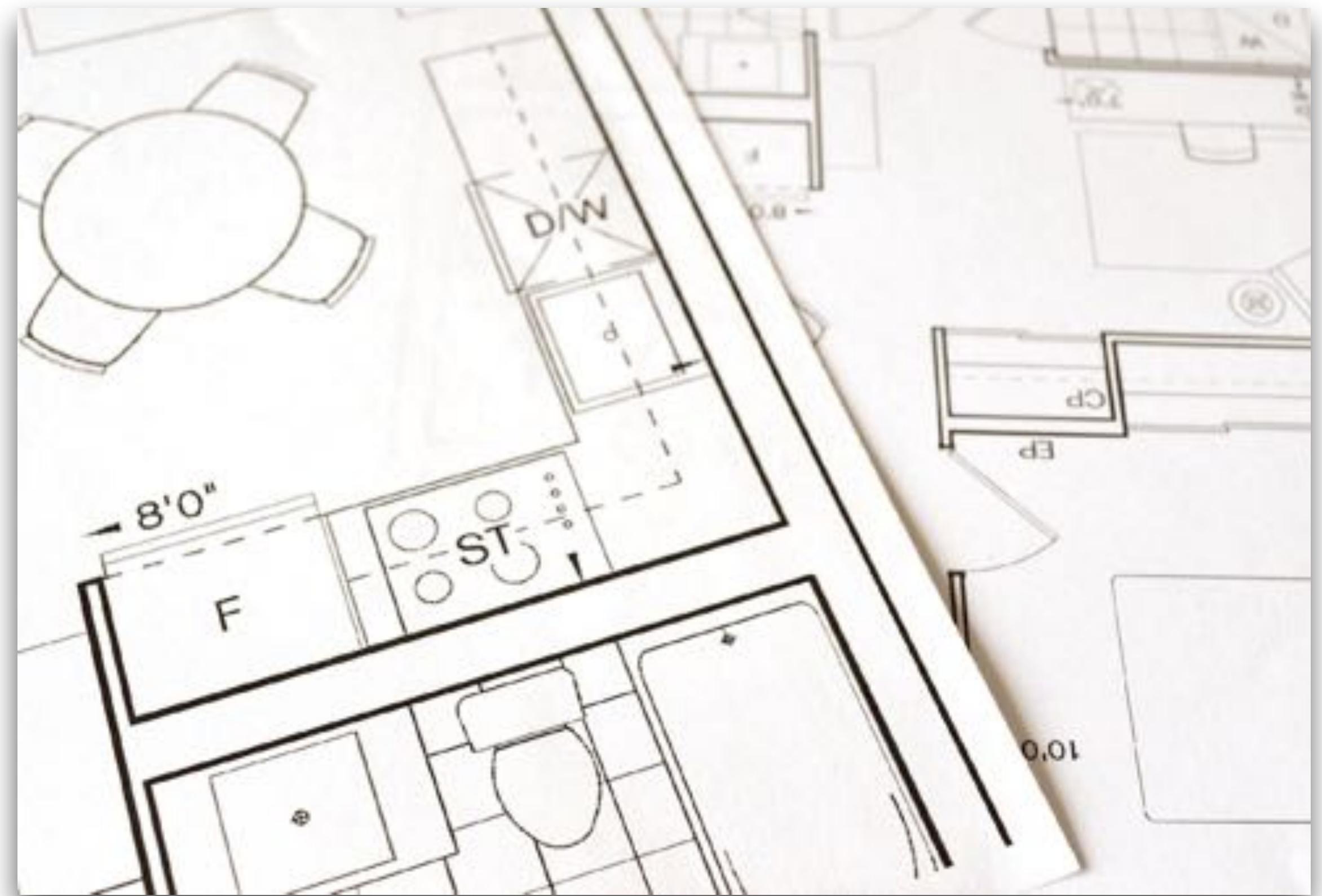
Sample environment requirements

- Run PHP applications
- Store information in a DB
- Upload and serve files



Sample environment requirements

- Run PHP applications
- Store information in a DB
- Upload and serve files
- Scale to handle many users with rare or nonexistent downtime



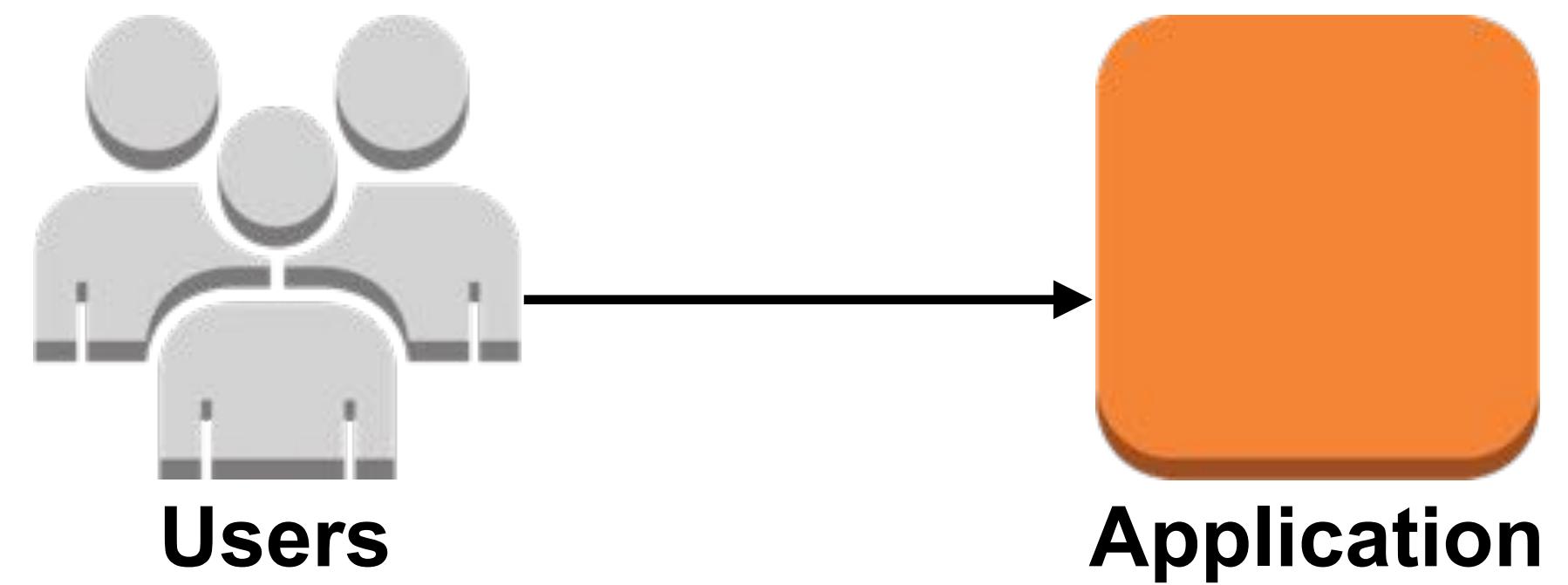
Elastic Compute Cloud (EC2)



Elastic Compute Cloud (EC2)



Elastic Compute Cloud (EC2)



Elastic Compute Cloud (EC2)



Launch Instance

Amazon Machine Image (AMI)



Amazon Machine Image (AMI)



- Template for launching EC2 instances



Amazon Machine Image (AMI)



- Template for launching EC2 instances
 - Operating system



Amazon Machine Image (AMI)



- Template for launching EC2 instances
 - Operating system
 - Applications



Amazon Machine Image (AMI)



- Template for launching EC2 instances
 - Operating system
 - Applications
 - Libraries



Amazon Machine Image (AMI)



- Template for launching EC2 instances
 - Operating system
 - Applications
 - Libraries
 - Configuration



Amazon Machine Image (AMI)



- Template for launching EC2 instances
 - Operating system
 - Applications
 - Libraries
 - Configuration
 - Data

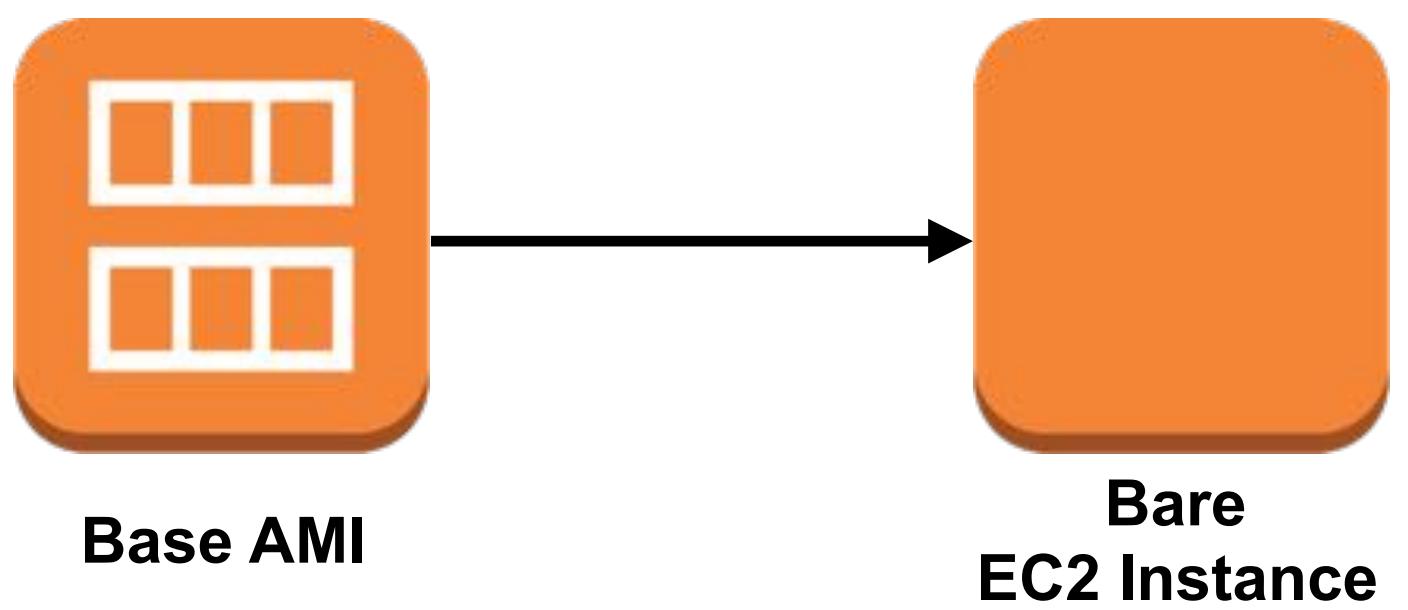


Amazon Machine Image (AMI)

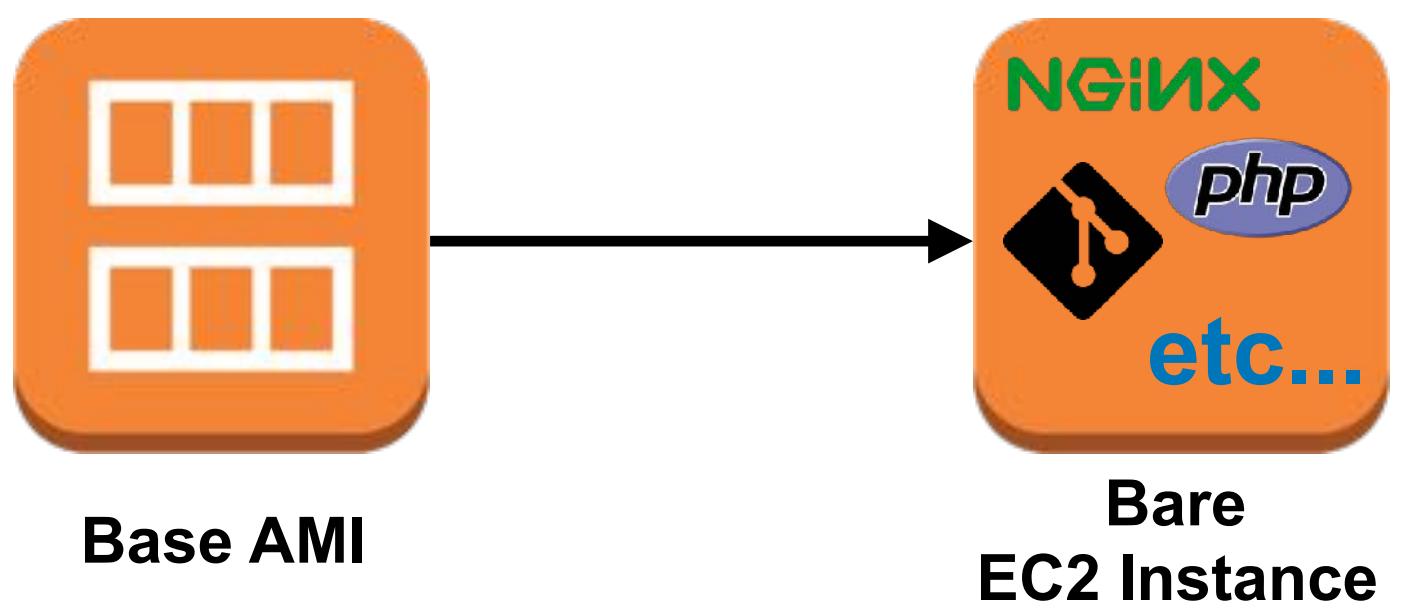


Base AMI

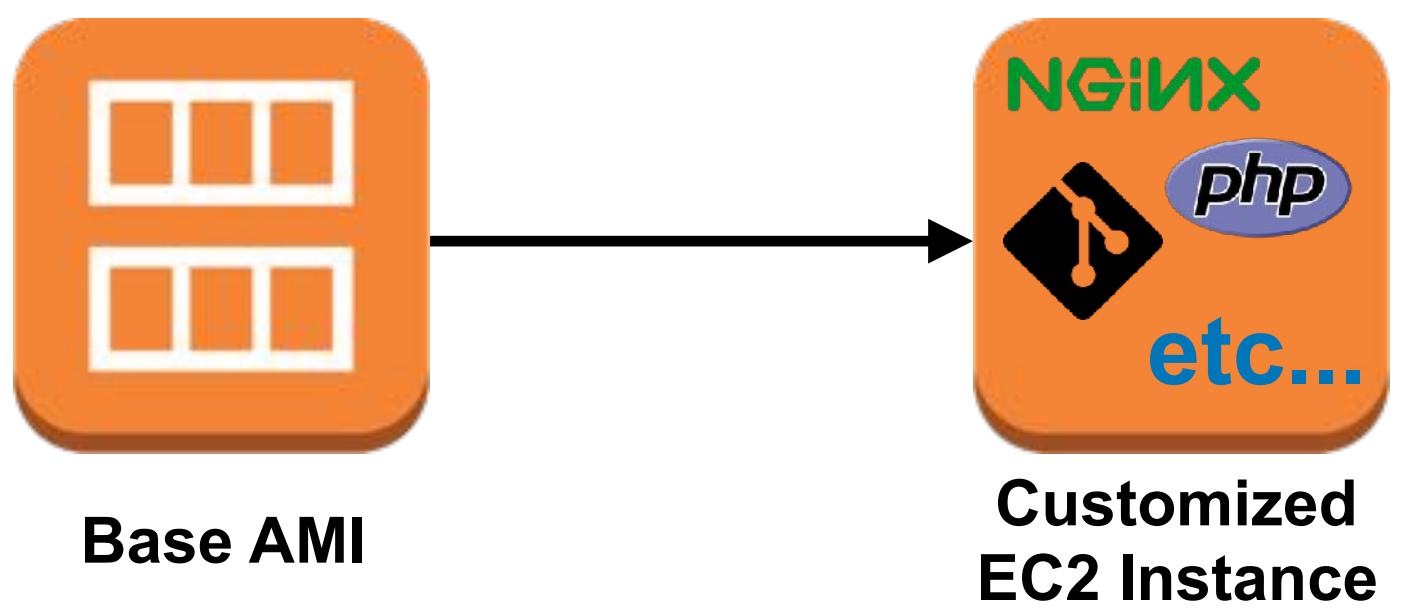
Amazon Machine Image (AMI)



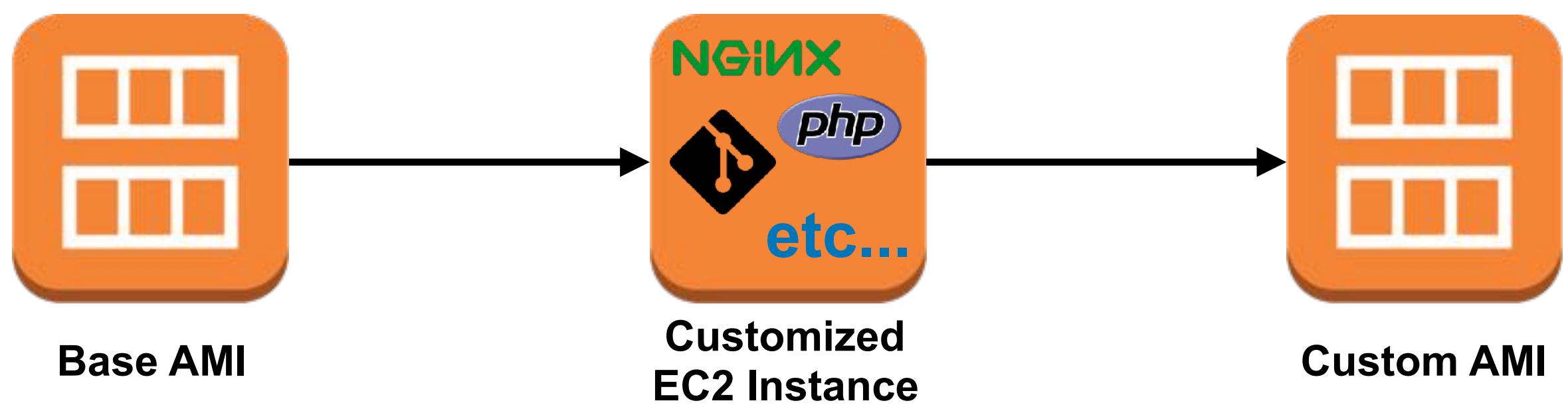
Amazon Machine Image (AMI)



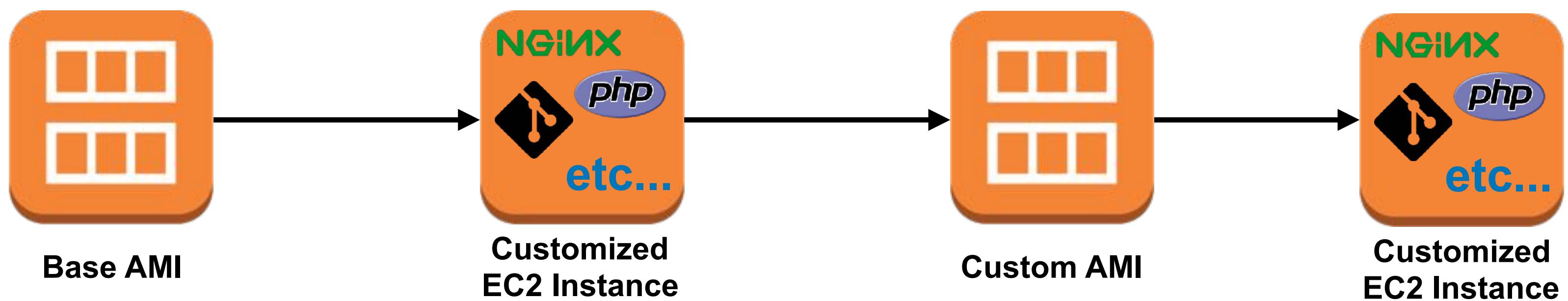
Amazon Machine Image (AMI)



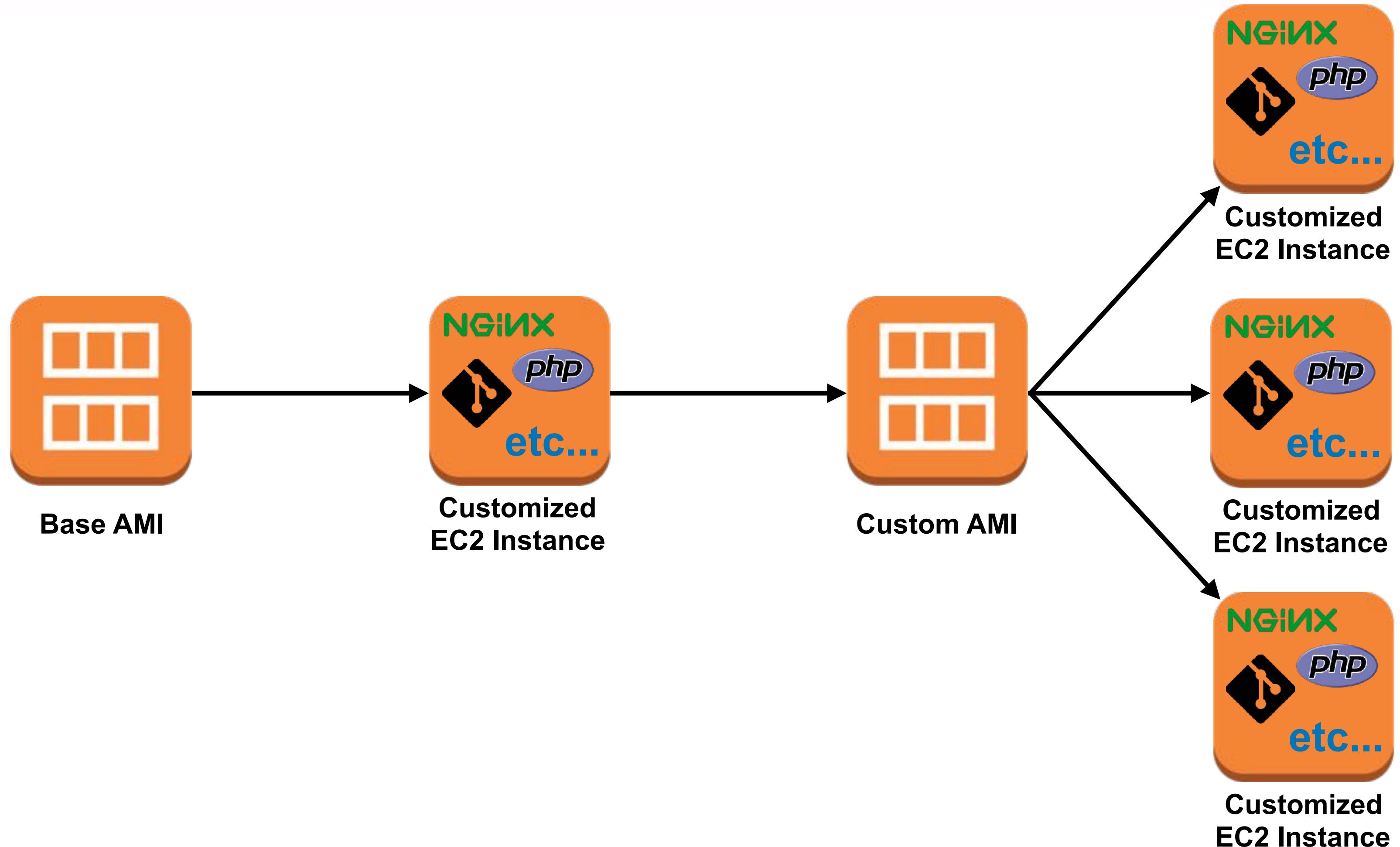
Amazon Machine Image (AMI)



Amazon Machine Image (AMI)



Amazon Machine Image (AMI)



EC2 Instance Classes



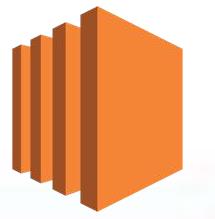
{family}{generation}.{size}

EC2 Instance Classes



c5.2xlarge

EC2 Instance Classes



c5.2xlarge

Compute-optimized

EC2 Instance Classes



c5.2xlarge

5th generation

EC2 Instance Classes



c5.2xlarge



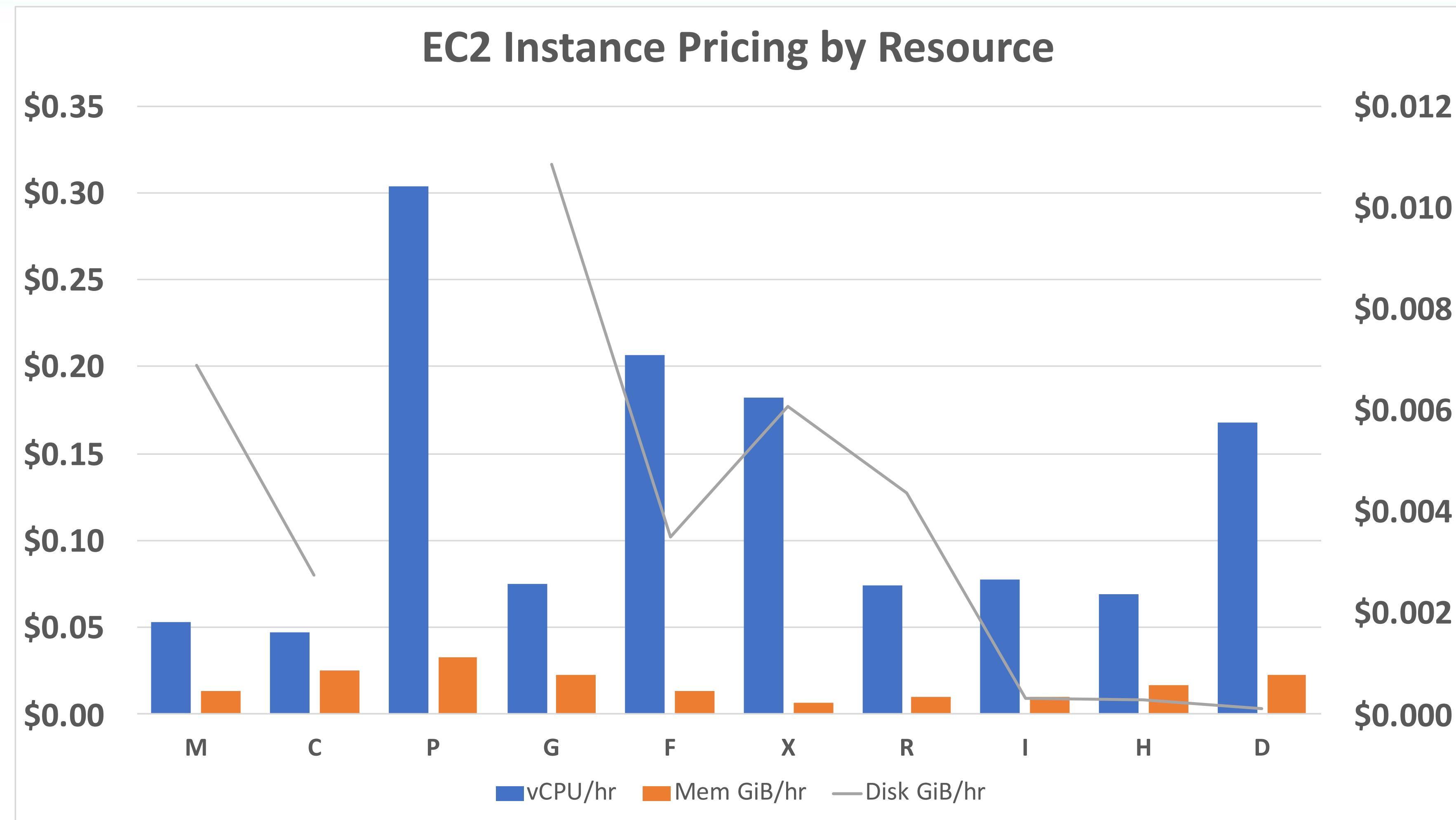
twice the size of c5.xlarge
(also, generally twice the price)

EC2 Instance Families

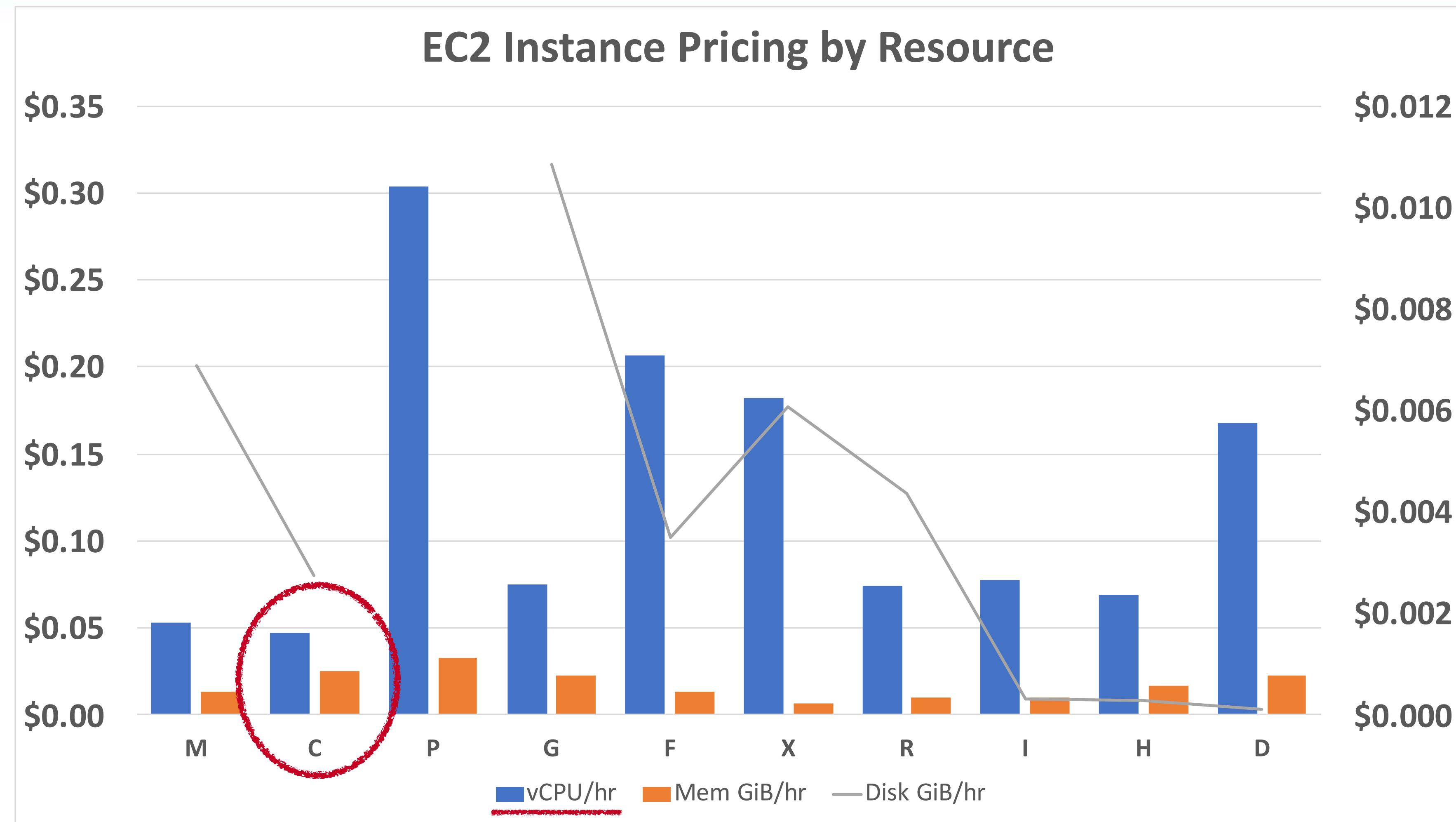


- General Purpose
- Compute Optimized
- Memory Optimized
- Storage Optimized
- Accelerated Computing (GPU)

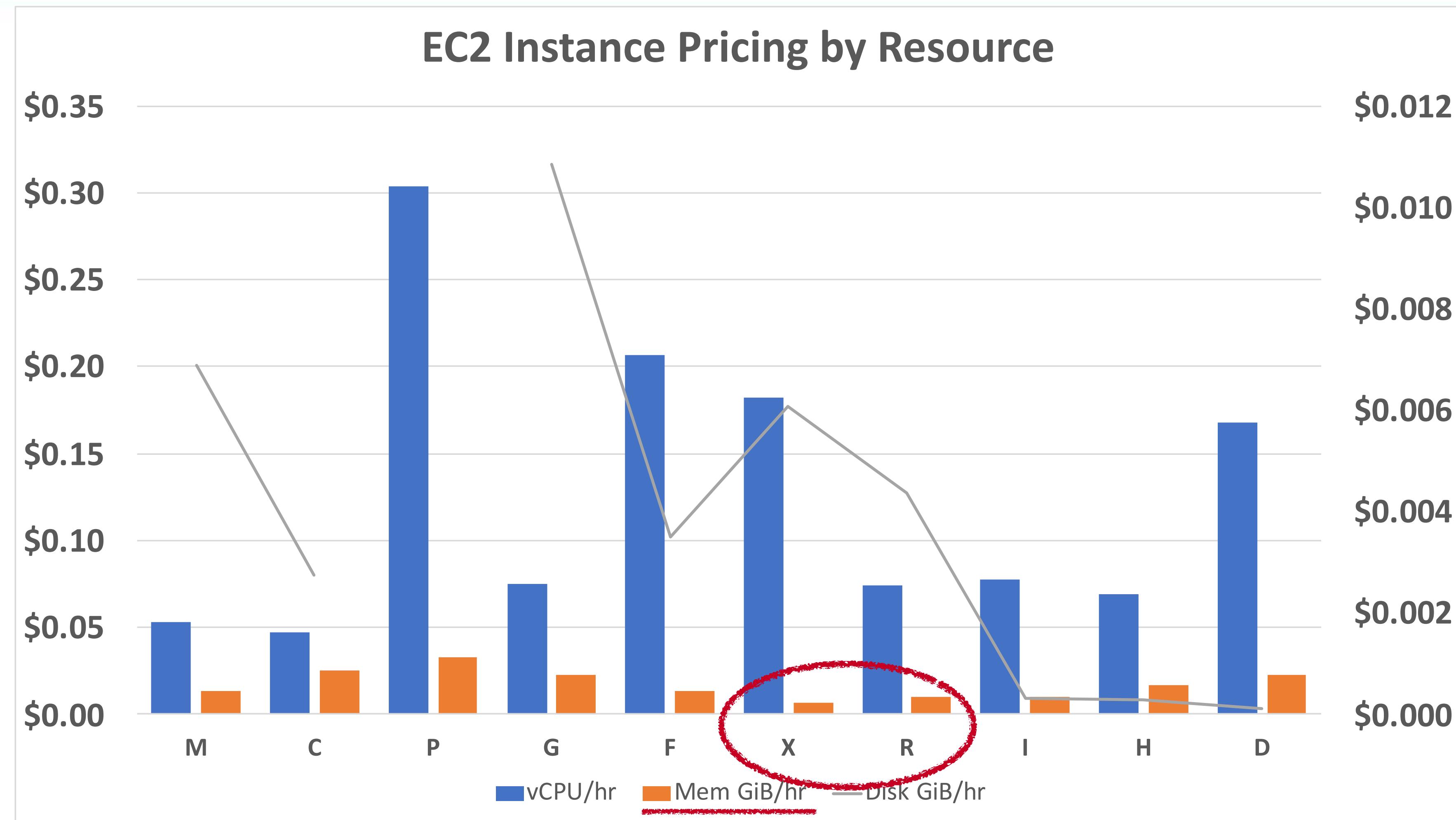
EC2 Instance Families



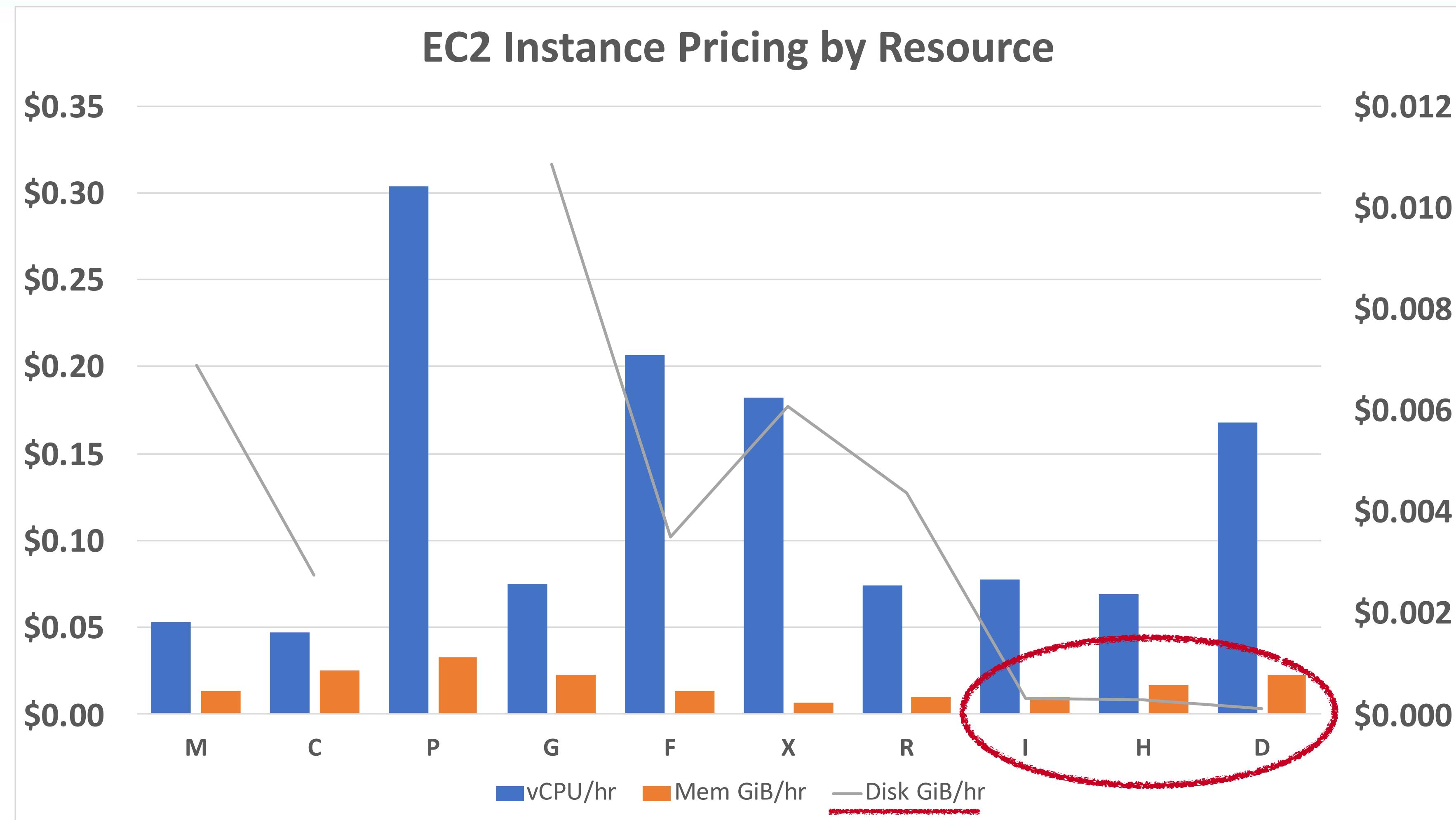
EC2 Instance Families



EC2 Instance Families



EC2 Instance Families



Elastic Compute Cloud (EC2)



Network



vpc-aded01d6 (default)



Elastic Compute Cloud (EC2)



?

Network i

vpc-aded01d6 (default) ▼

A screenshot of an AWS EC2 instance configuration screen. The 'Network' section is selected. A red circle highlights the VPC dropdown menu, which shows 'vpc-aded01d6 (default)'. A red question mark is positioned above the network section.

Virtual Private Cloud (VPC)



Virtual Private Cloud (VPC)

- Isolated virtual network within AWS



Virtual Private Cloud (VPC)

- Isolated virtual network within AWS
- Fully customizable to your needs:



Virtual Private Cloud (VPC)

- Isolated virtual network within AWS
- Fully customizable to your needs:
 - IP address range, subnets, routing



Virtual Private Cloud (VPC)

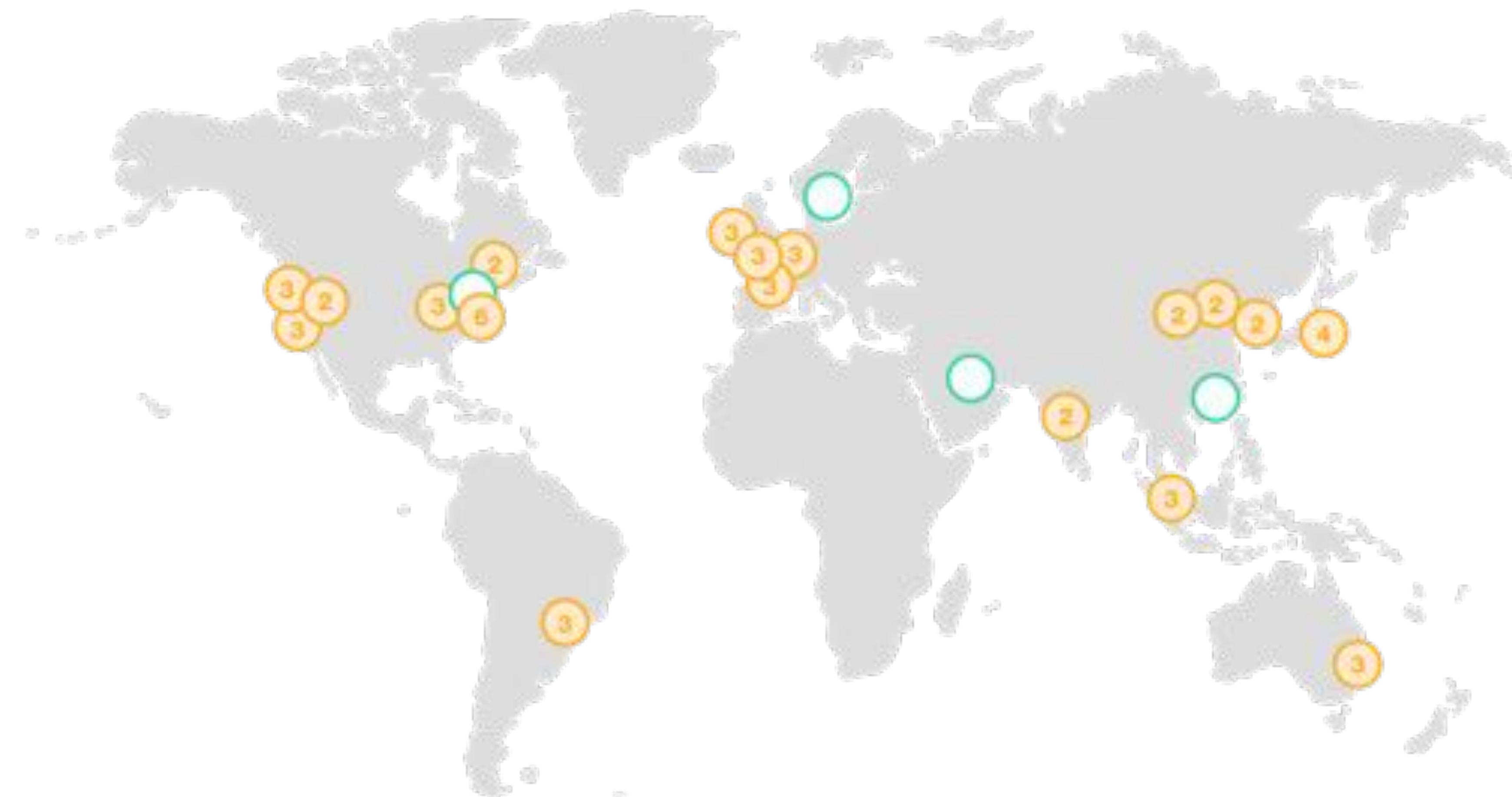
- Isolated virtual network within AWS
- Fully customizable to your needs:
 - IP address range, subnets, routing
 - Security groups, network access control lists, gateways, etc.



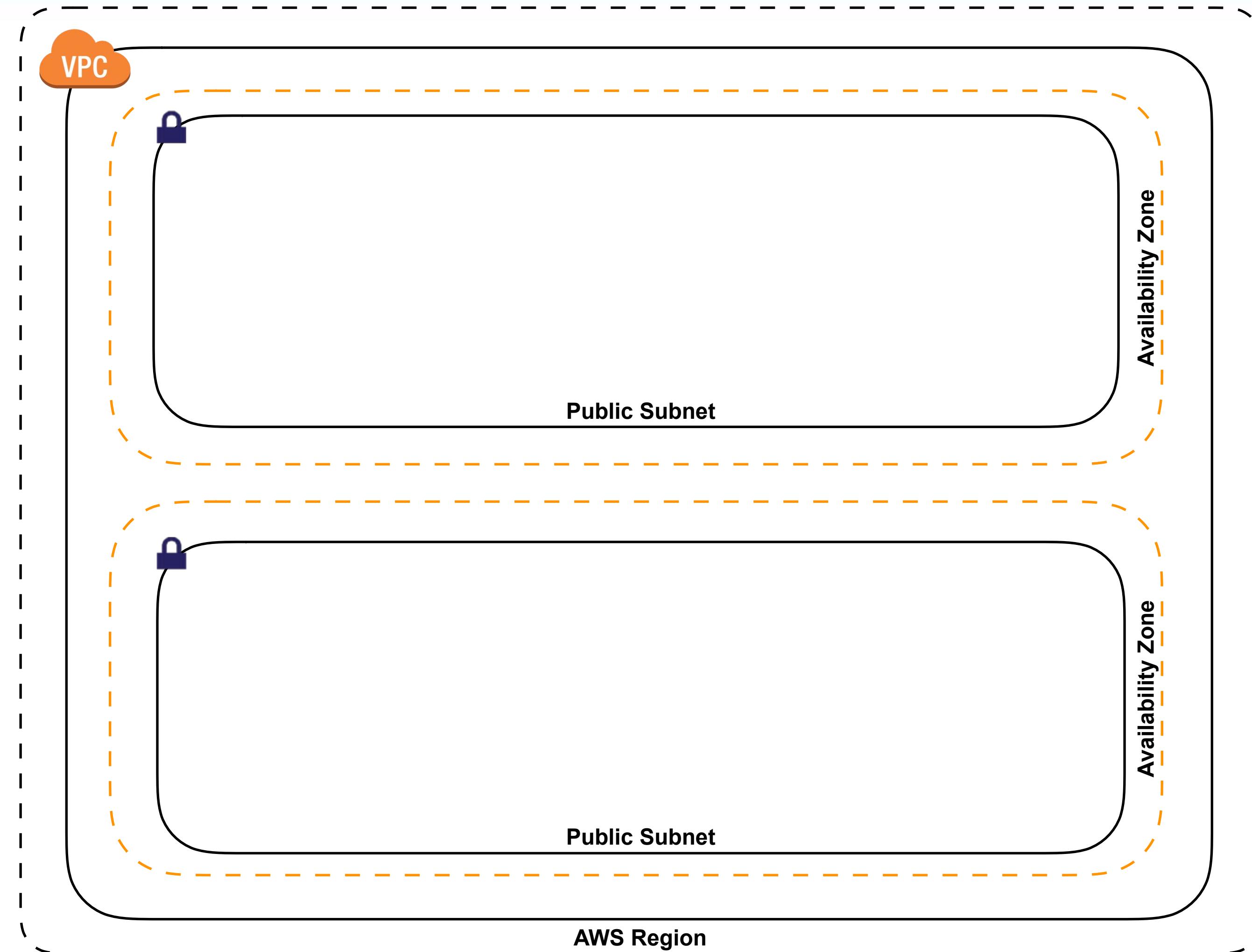
Virtual Private Cloud (VPC)



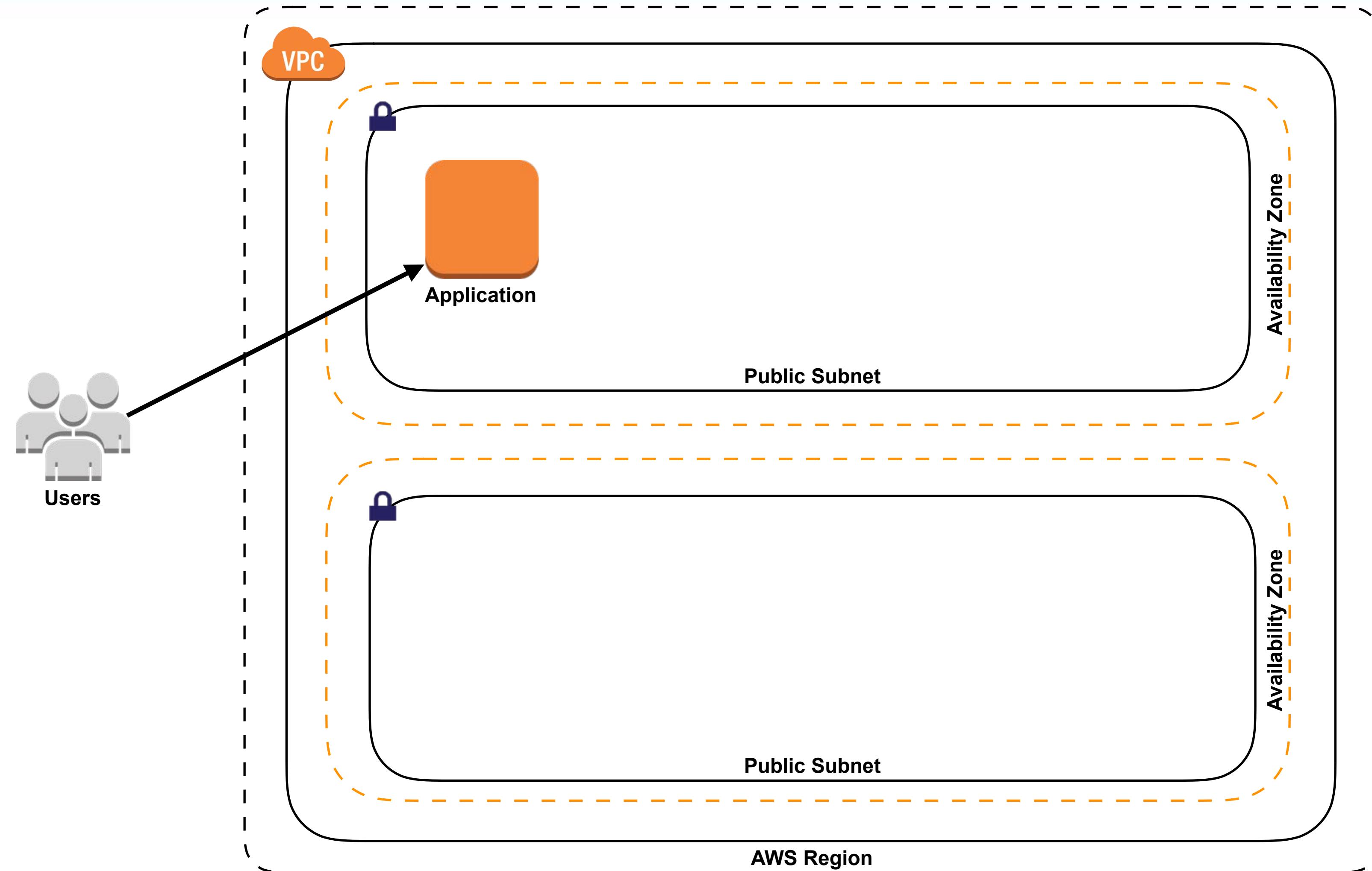
50+ Availability Zones across 18 Regions



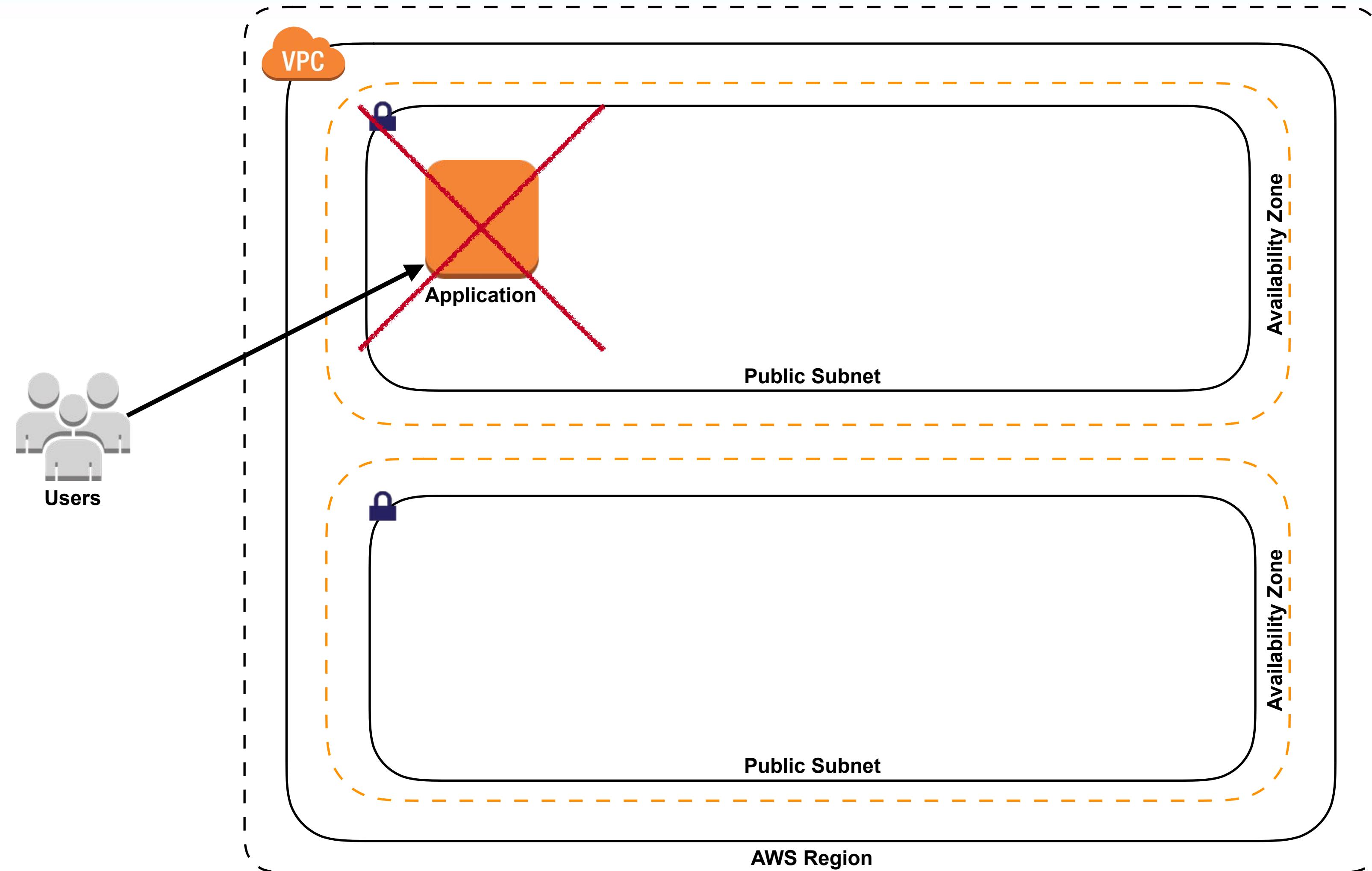
Virtual Private Cloud (VPC)



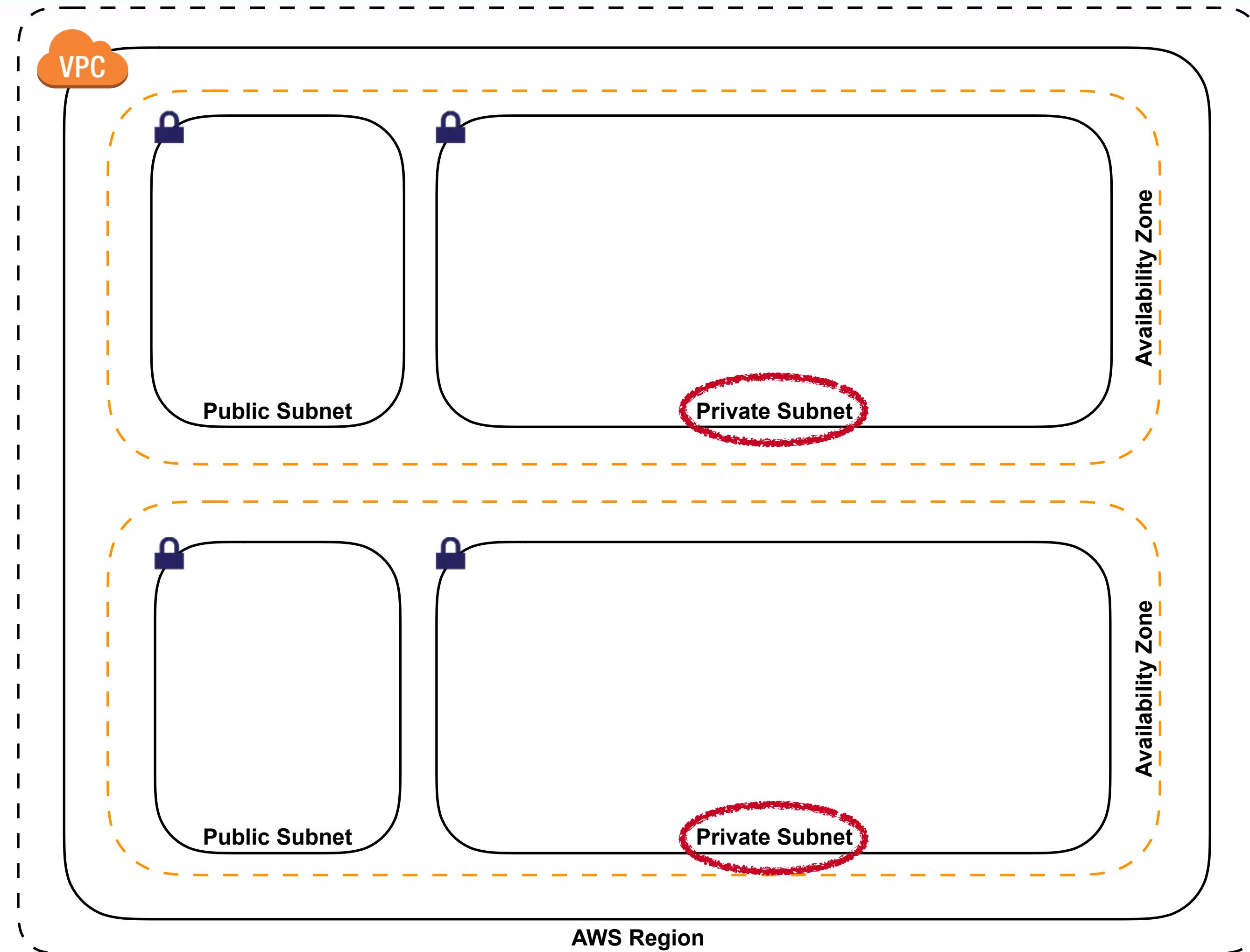
Virtual Private Cloud (VPC)



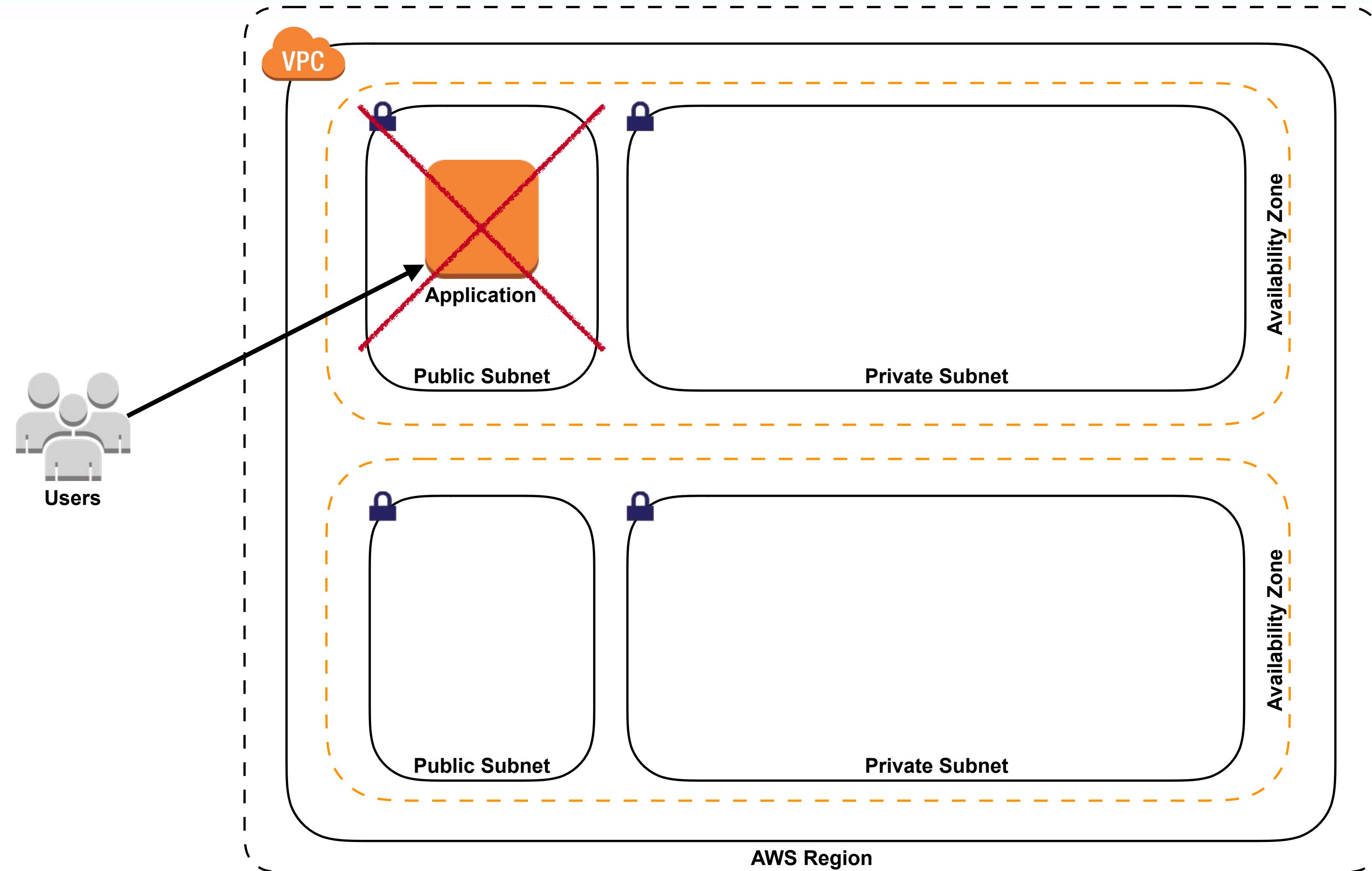
Virtual Private Cloud (VPC)



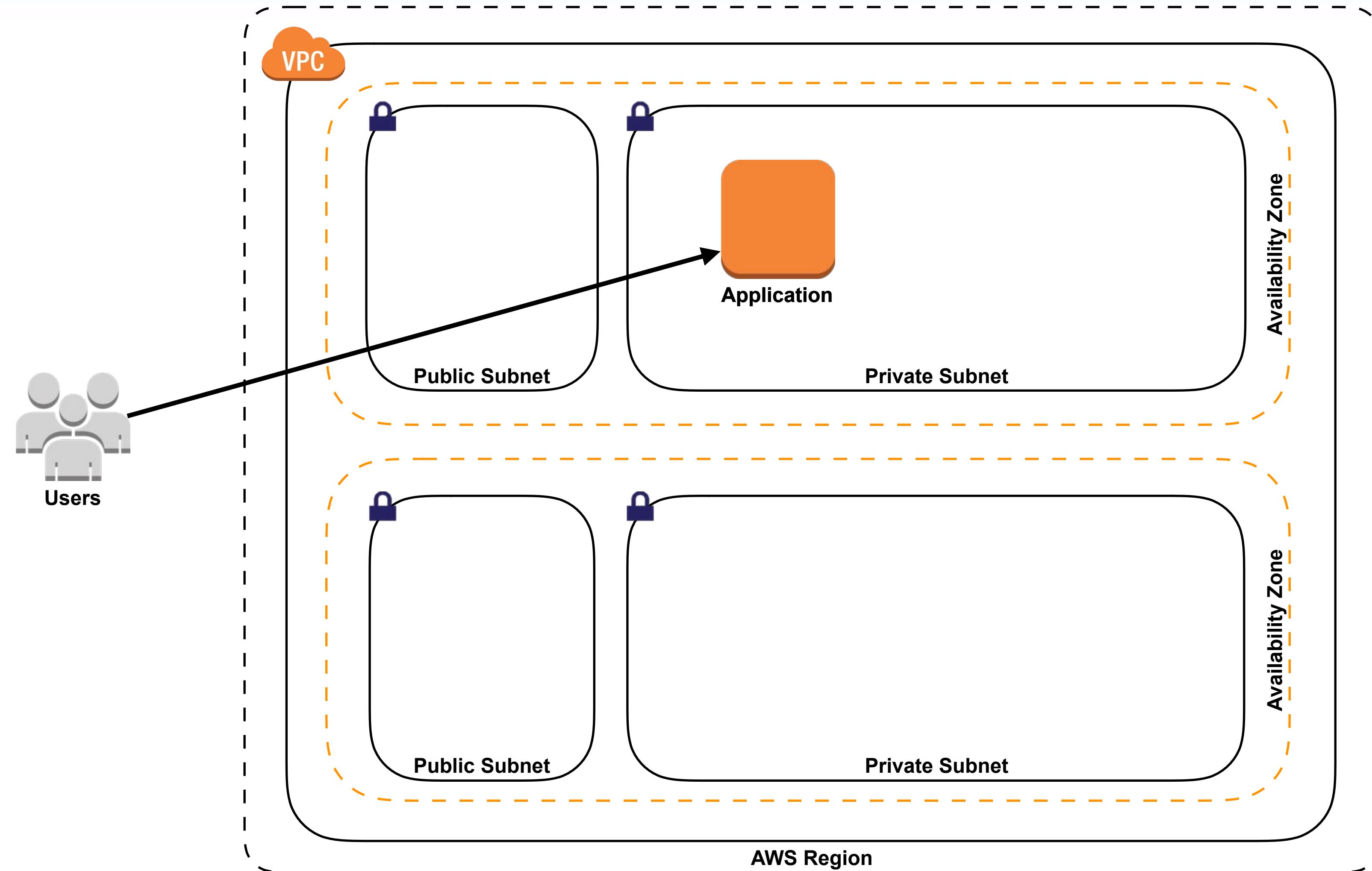
Virtual Private Cloud (VPC)



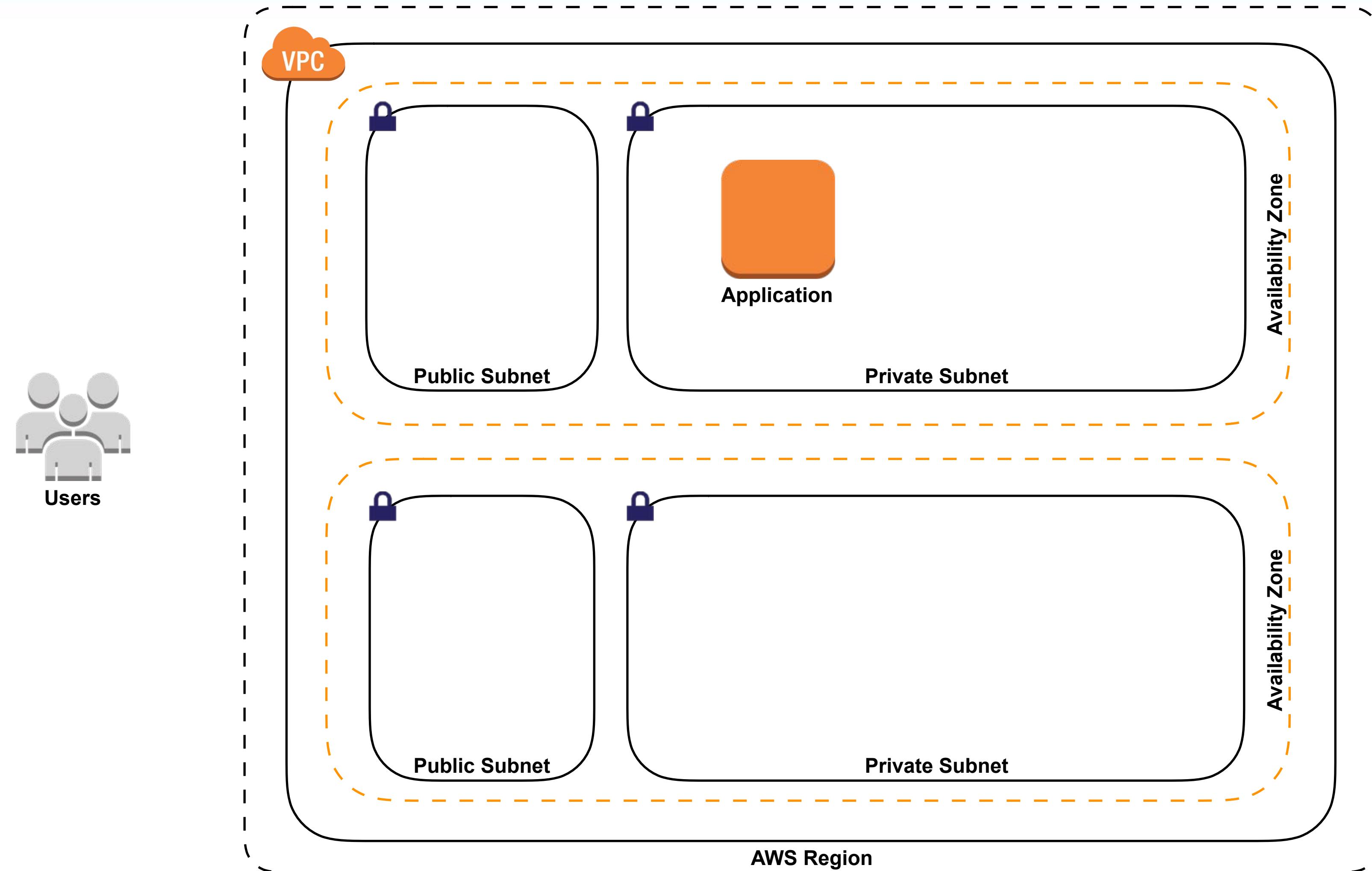
Virtual Private Cloud (VPC)



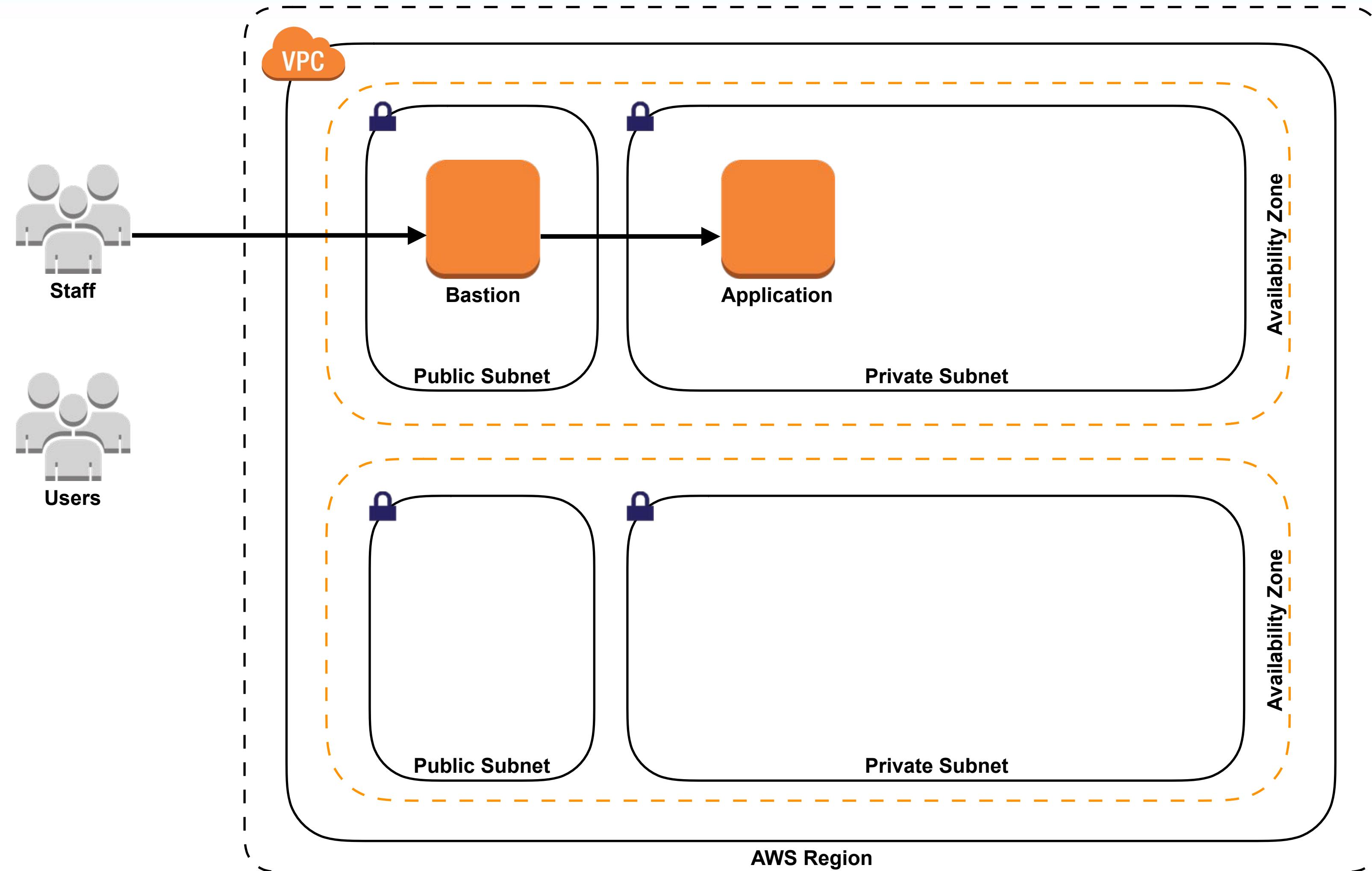
Virtual Private Cloud (VPC)



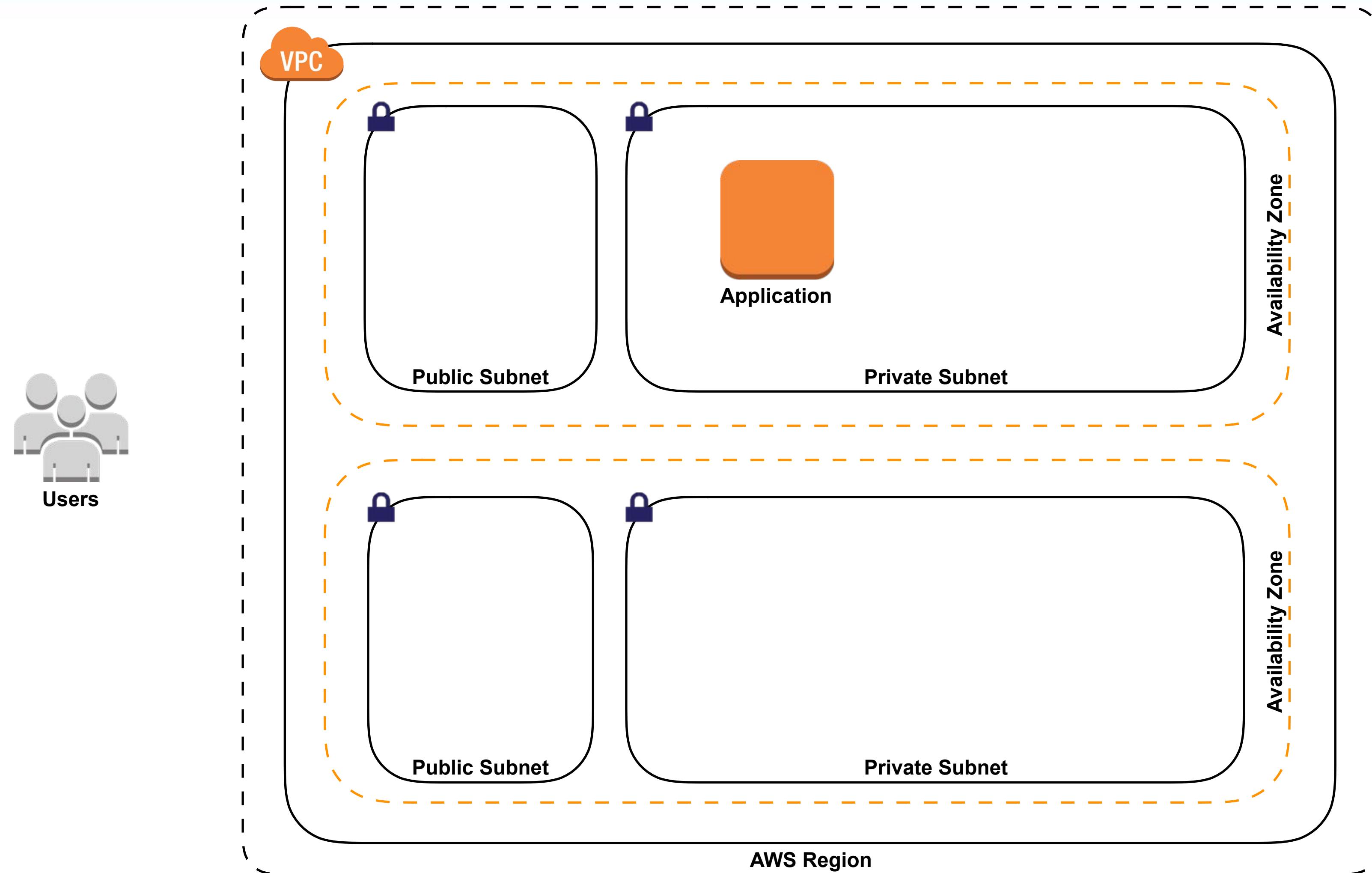
Virtual Private Cloud (VPC)



Virtual Private Cloud (VPC)



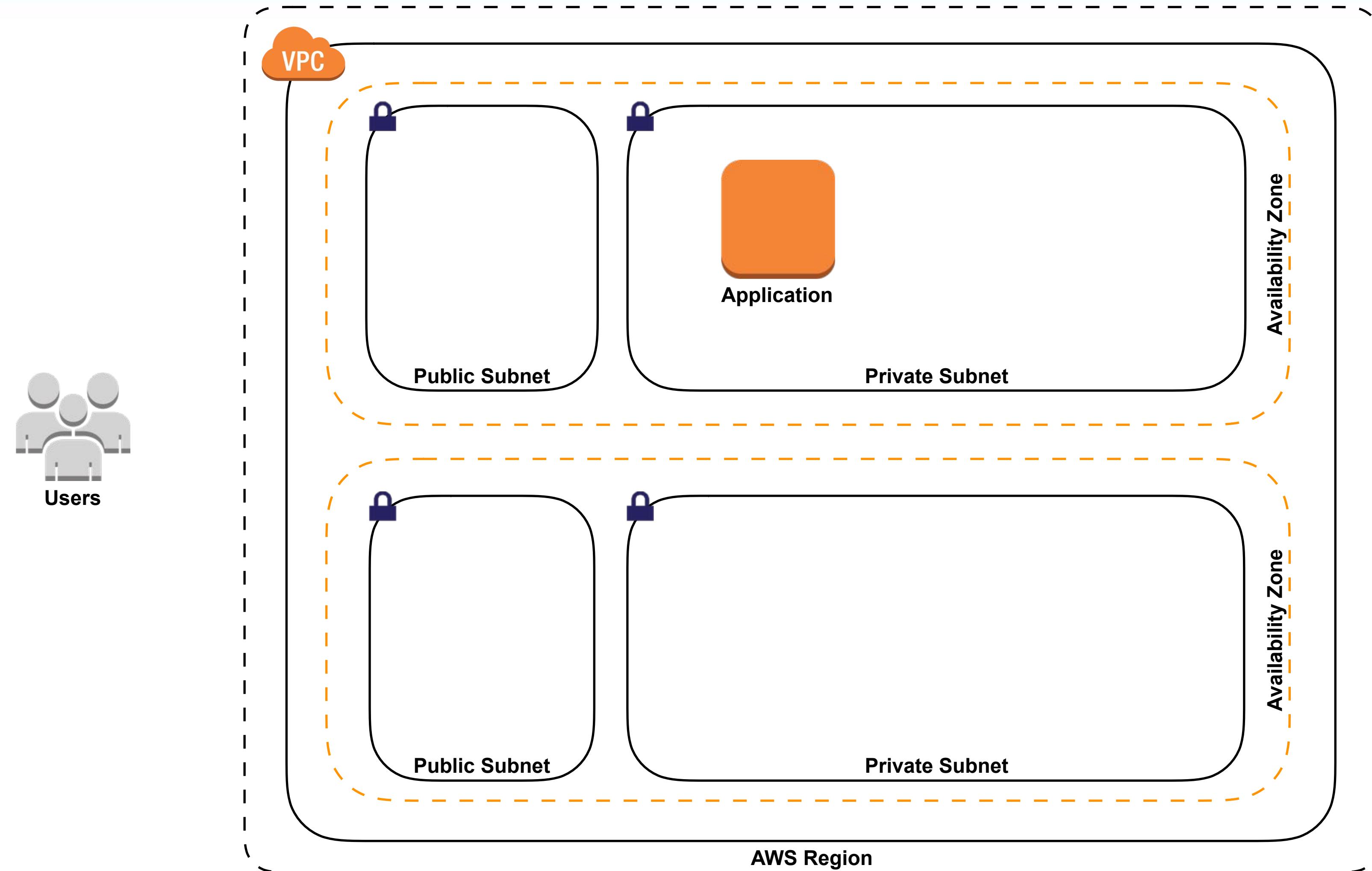
Virtual Private Cloud (VPC)





Security is about layers

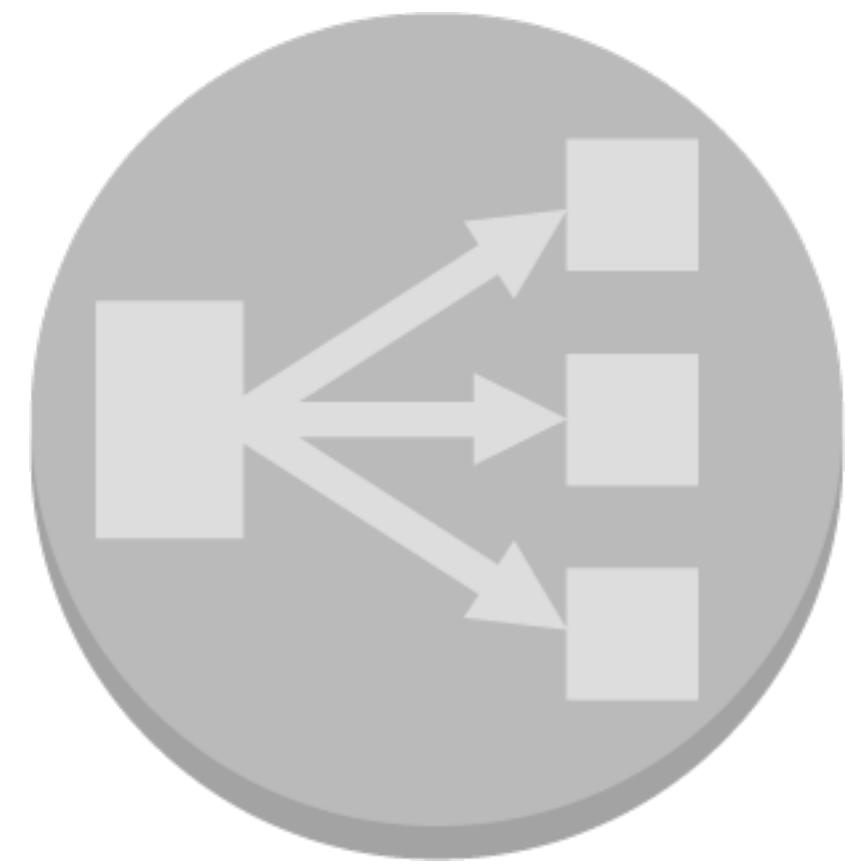
Virtual Private Cloud (VPC)



Elastic Load Balancing (ELB)



Elastic Load Balancing (ELB)



Classic Load Balancer

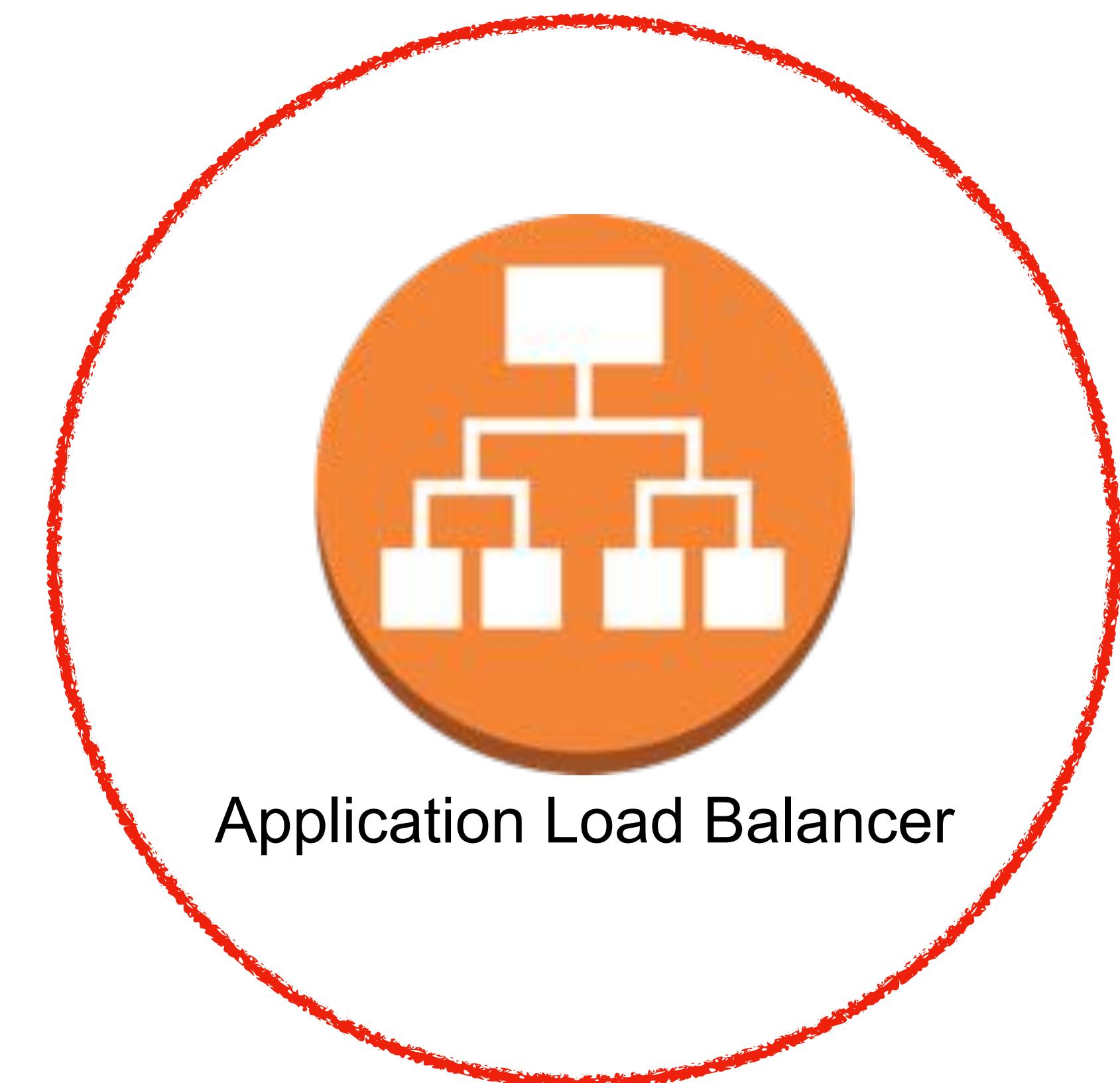


Application Load Balancer



Network Load Balancer

Elastic Load Balancing (ELB)



Application Load Balancer (ALB)



- Single source of user traffic



Application Load Balancer (ALB)



- Single source of user traffic
- Listen for requests and distribute to registered targets



Application Load Balancer (ALB)



- Single source of user traffic
- Listen for requests and distribute to registered targets
- Scales to meet demand



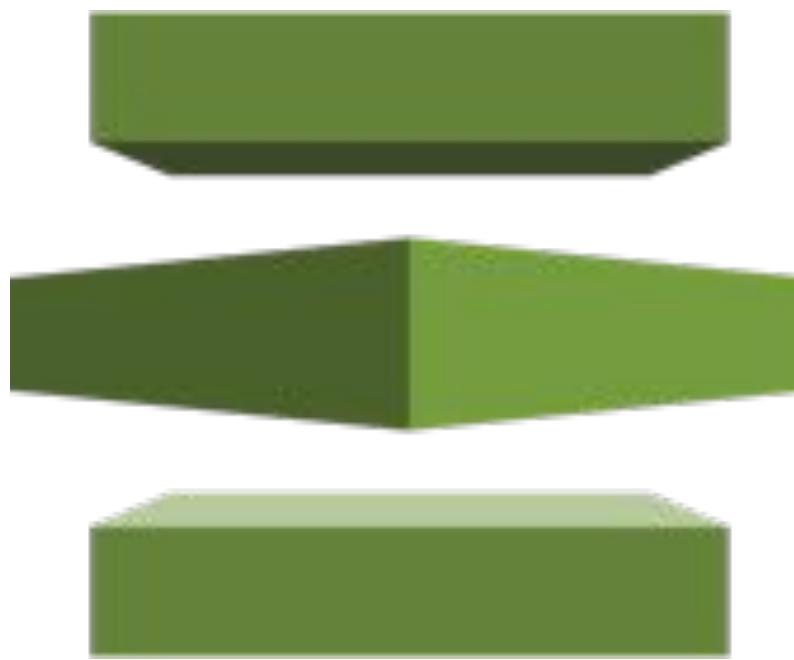
Application Load Balancer (ALB)



- Single source of user traffic
- Listen for requests and distribute to registered targets
- Scales to meet demand
- SSL/TLS support



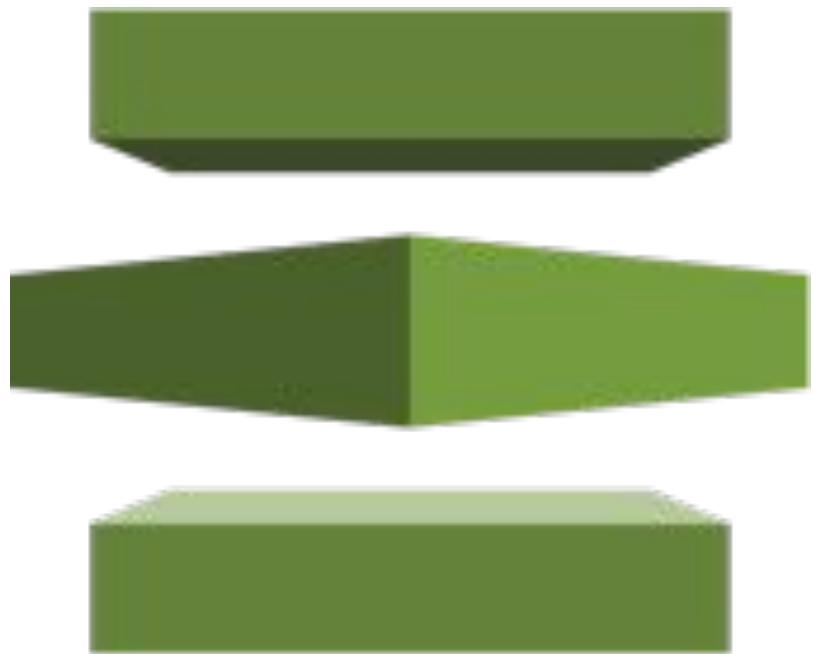
AWS Certificate Manager (ACM)



AWS Certificate Manager (ACM)



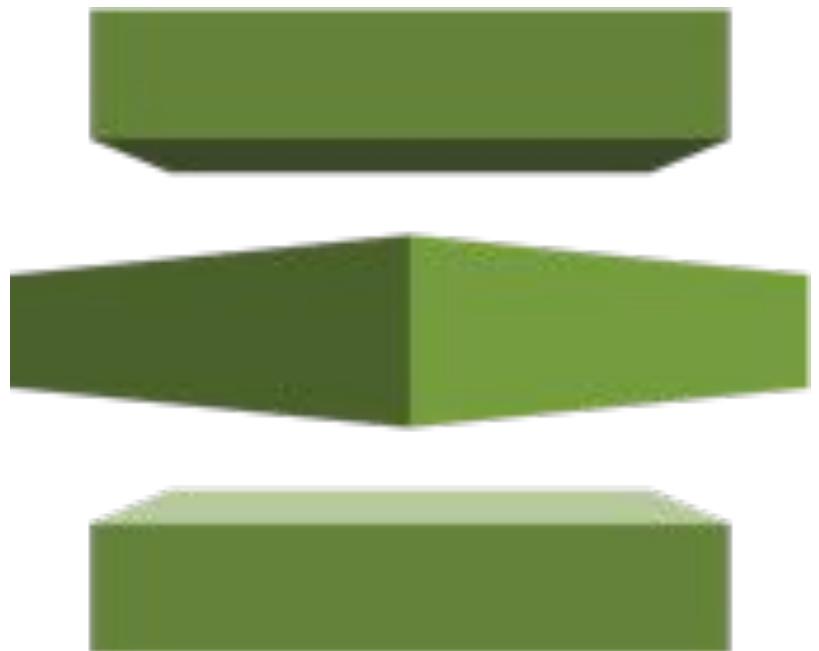
- SSL/TLS certificates for AWS services



AWS Certificate Manager (ACM)



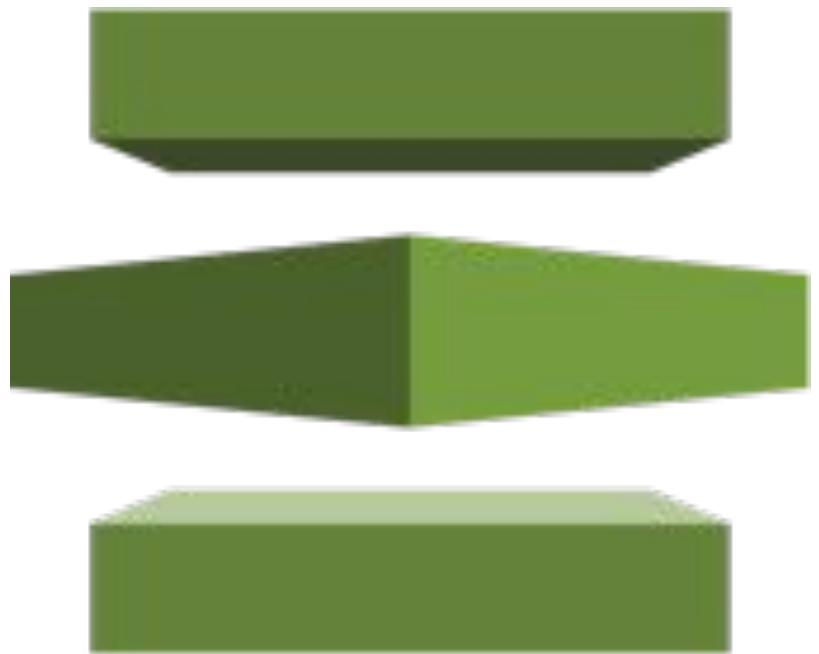
- SSL/TLS certificates for AWS services
- Create your own in seconds



AWS Certificate Manager (ACM)



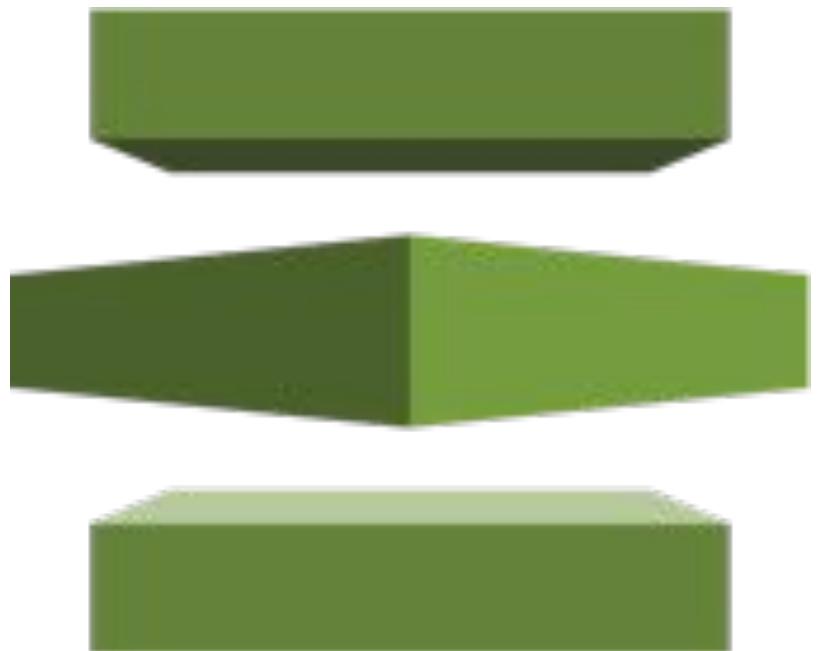
- SSL/TLS certificates for AWS services
- Create your own in seconds
- Easy to use



AWS Certificate Manager (ACM)



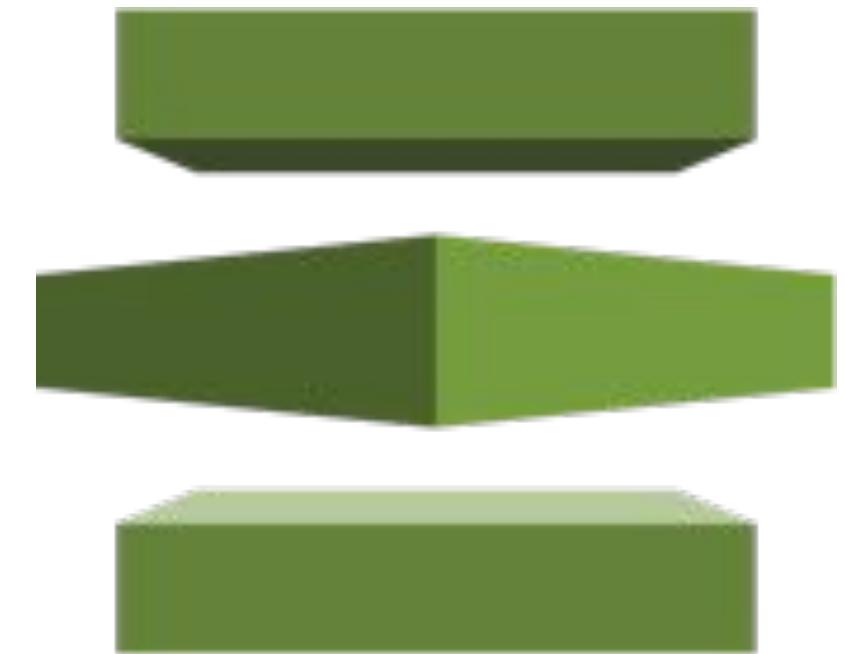
- SSL/TLS certificates for AWS services
- Create your own in seconds
- Easy to use
- Automatically renewed



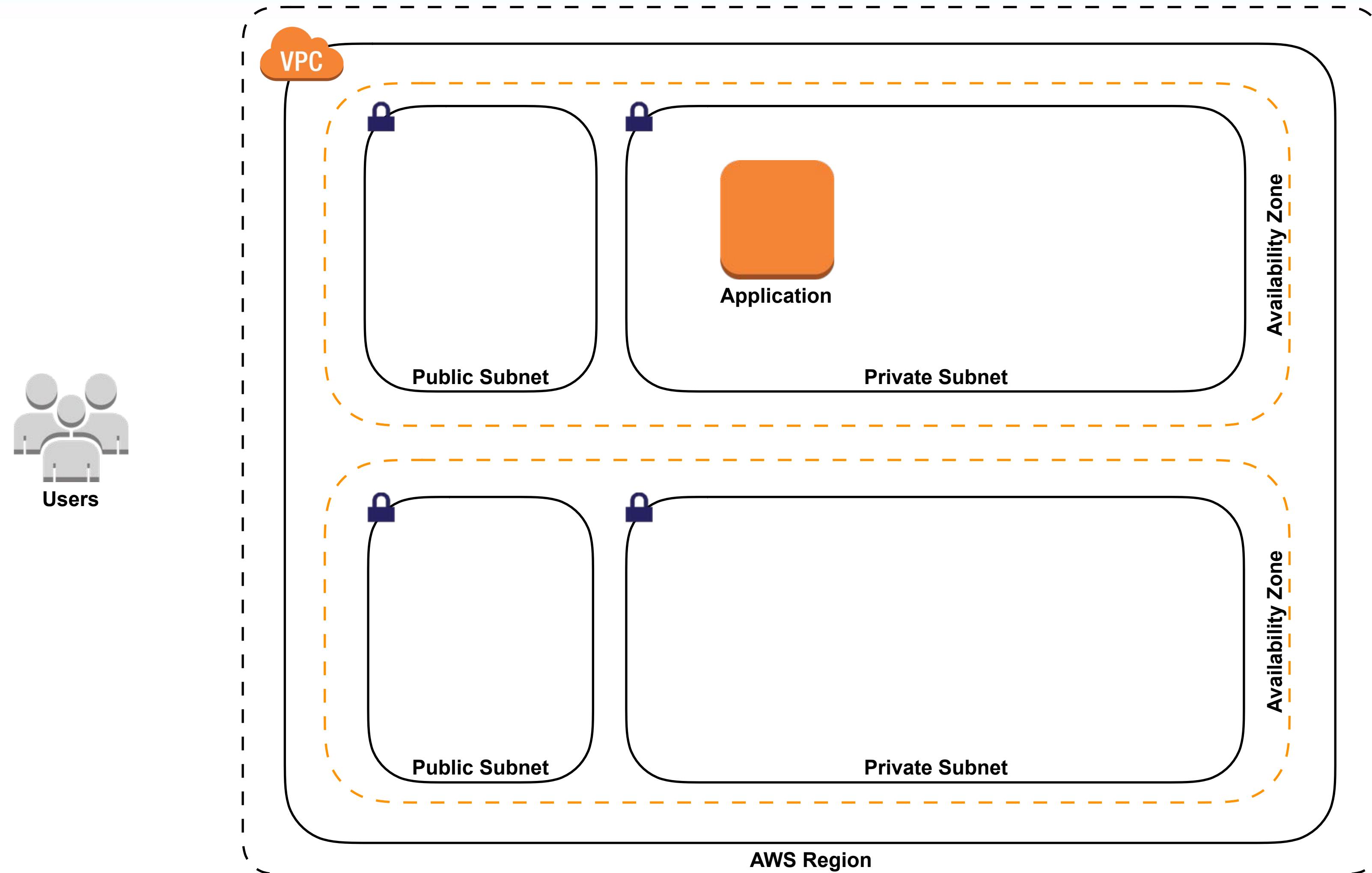
AWS Certificate Manager (ACM)



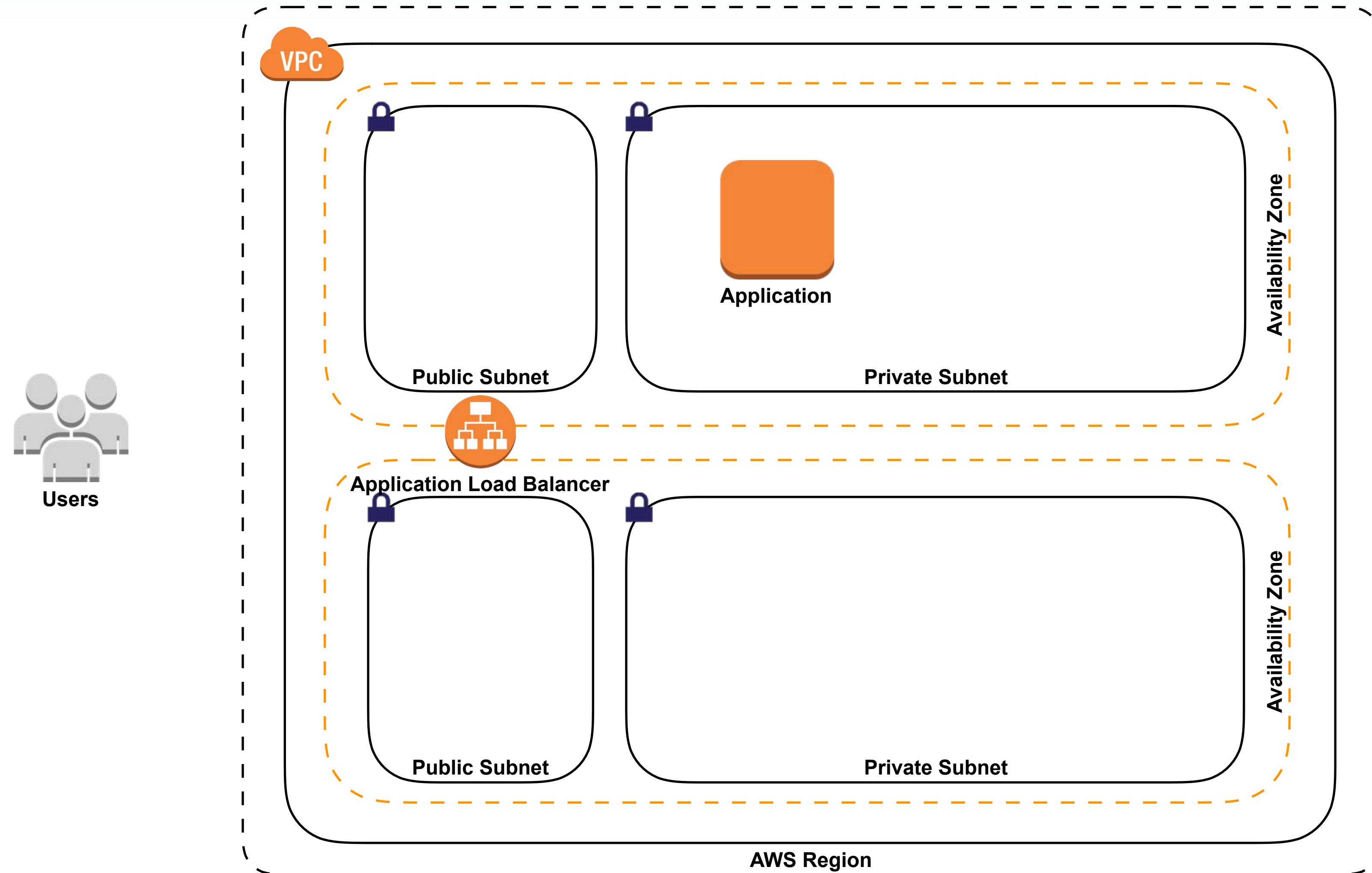
- SSL/TLS certificates for AWS services
- Create your own in seconds
- Easy to use
- Automatically renewed
- Free!



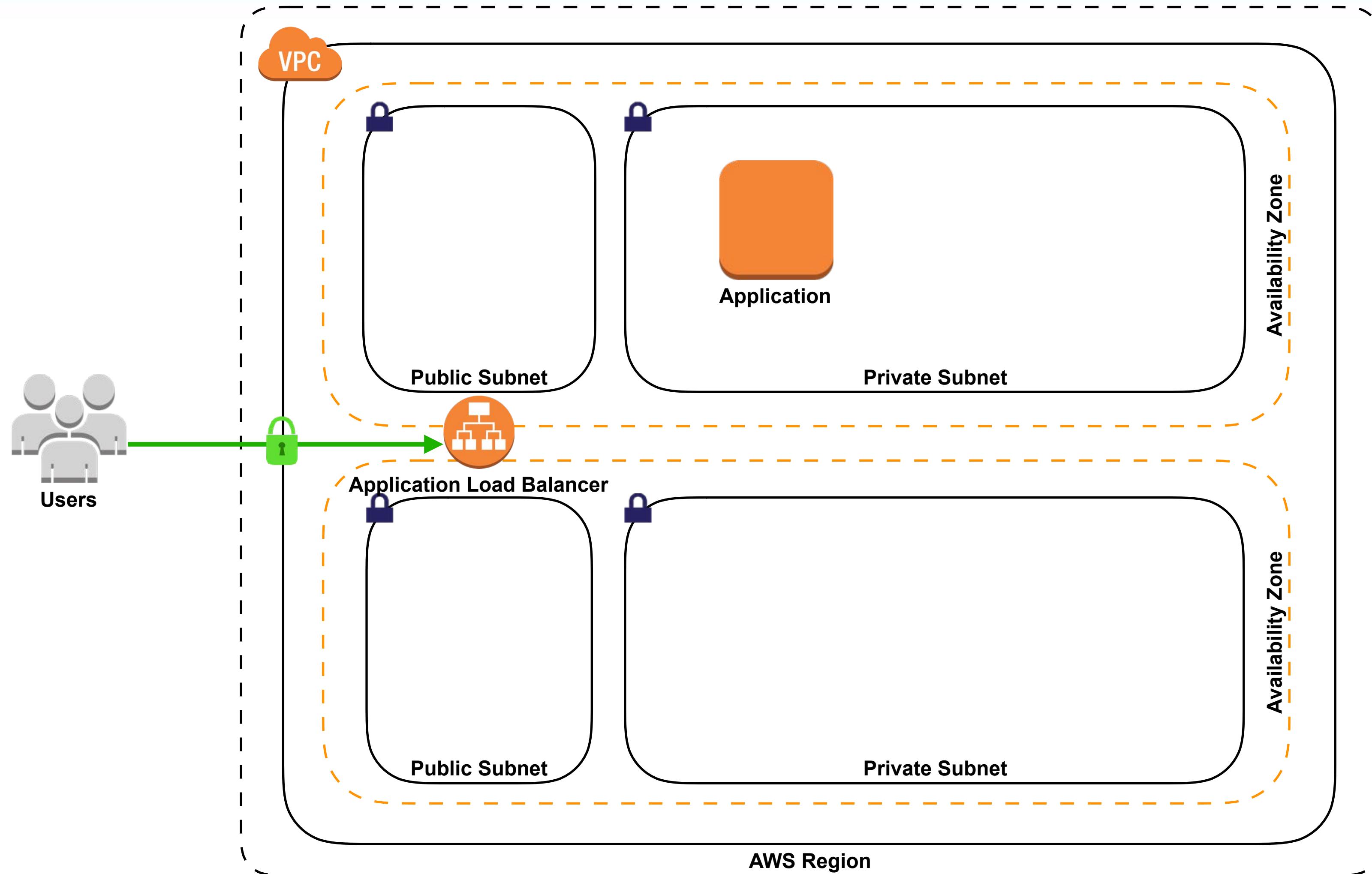
Application Load Balancer (ALB)



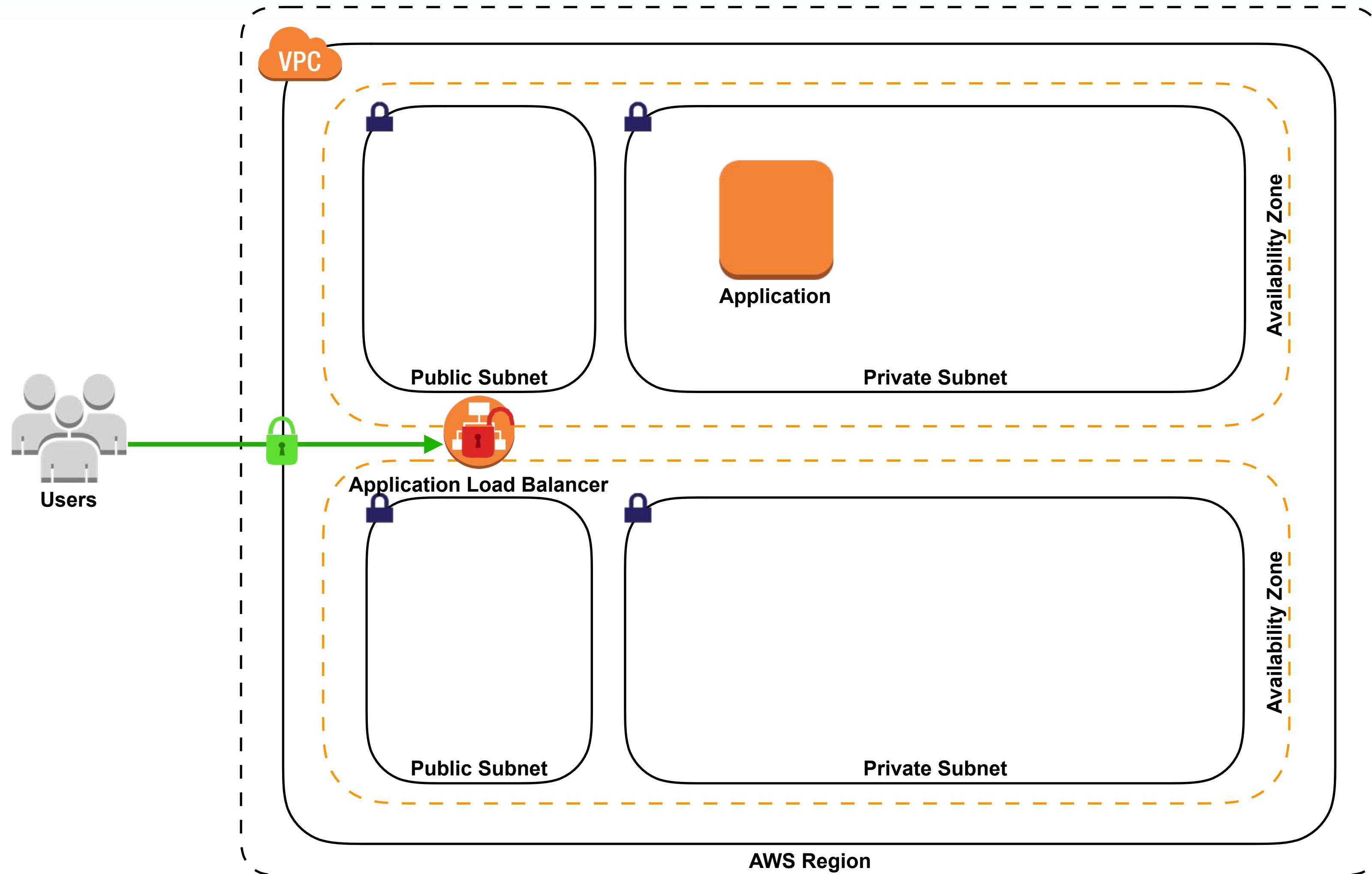
Application Load Balancer (ALB)



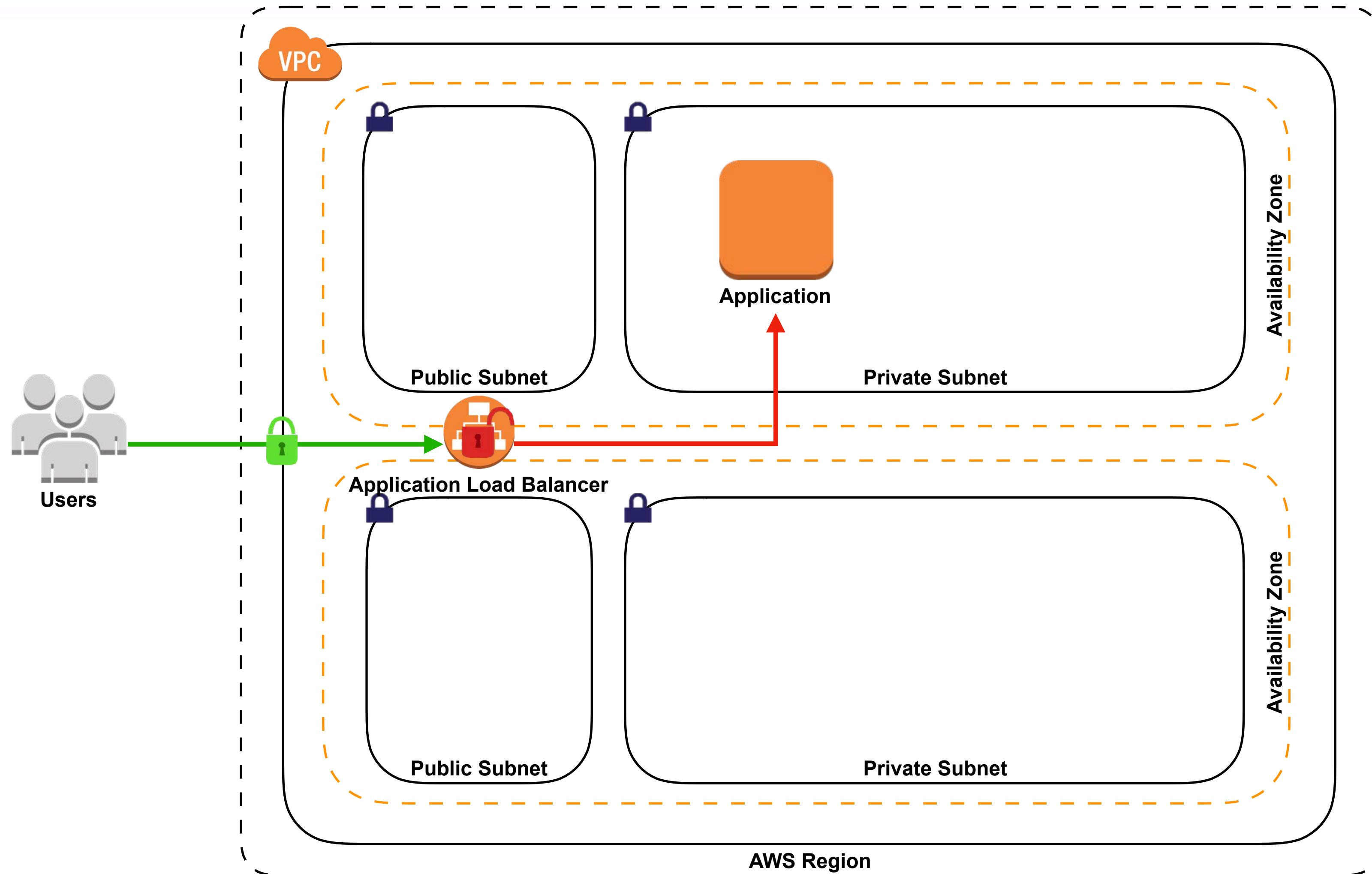
Application Load Balancer (ALB)



Application Load Balancer (ALB)



Application Load Balancer (ALB)



Amazon Route 53



Amazon Route 53

- Highly-available, reliable DNS service



Amazon Route 53

- Highly-available, reliable DNS service
- Public and private zones



Amazon Route 53

- Highly-available, reliable DNS service
- Public and private zones
- **Flexible routing**

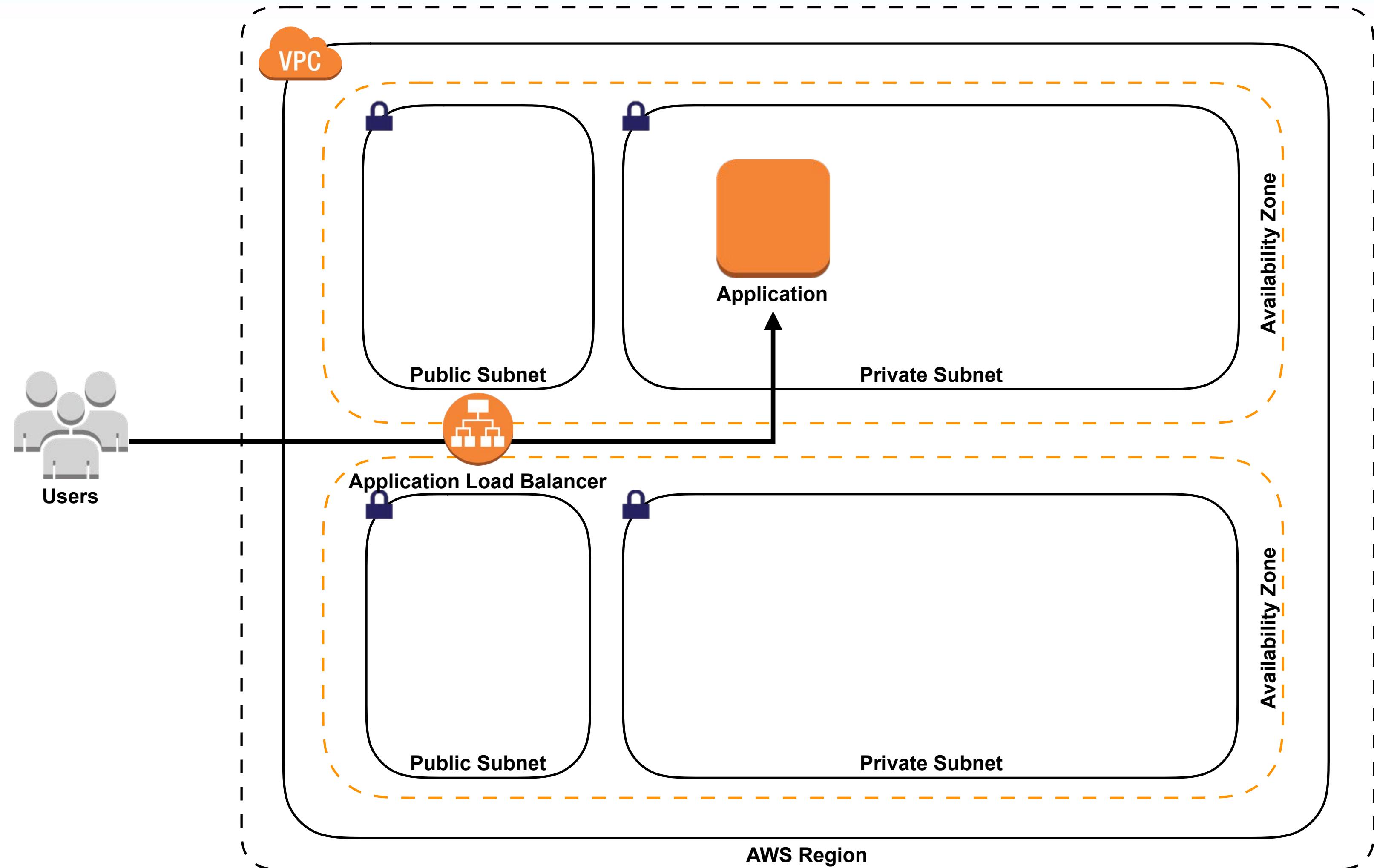


Amazon Route 53

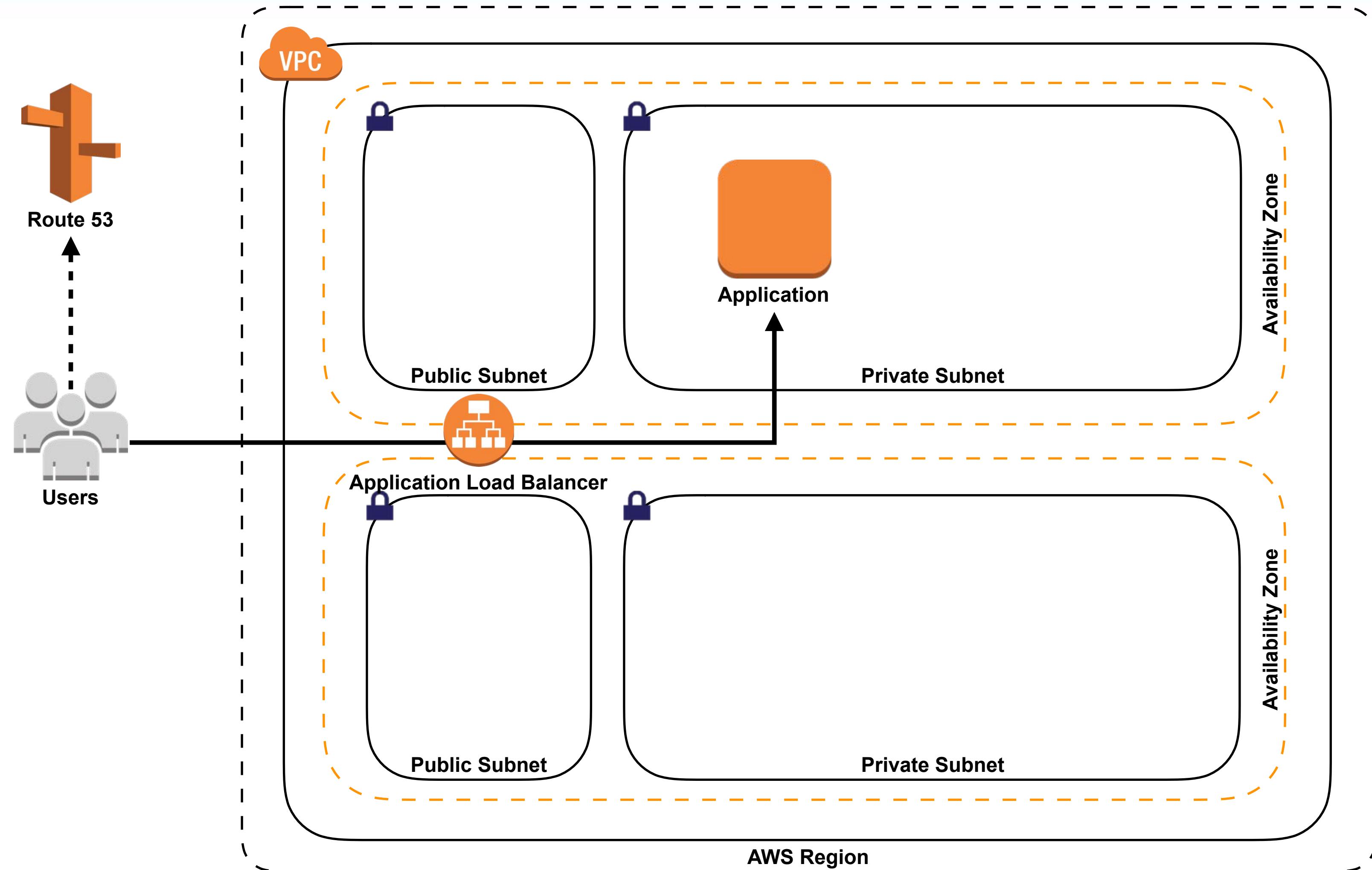
- Highly-available, reliable DNS service
- Public and private zones
- Flexible routing
- Well-integrated with other service offerings on AWS



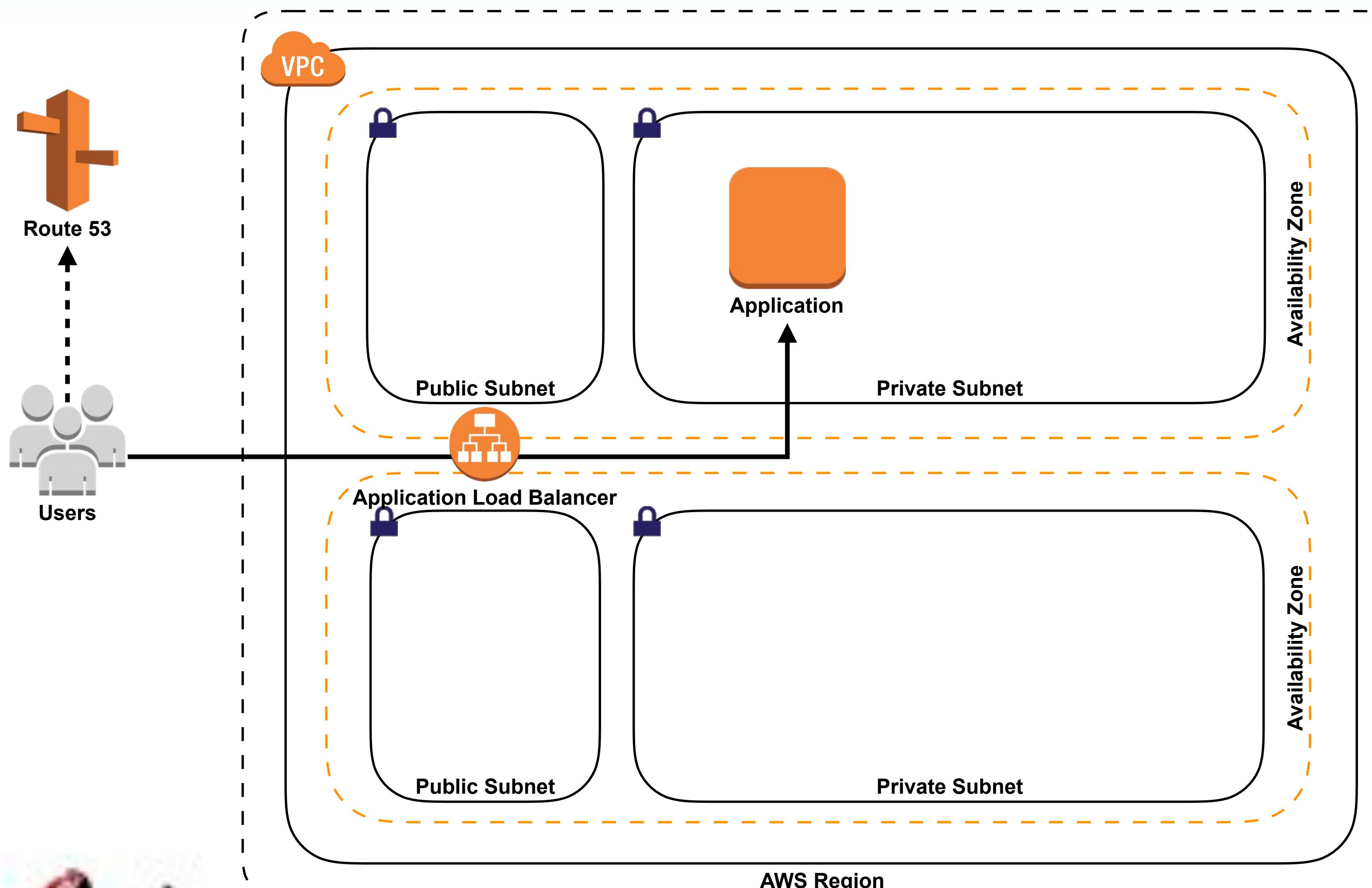
Amazon Route 53



Amazon Route 53



Checkpoint



- Highly-available
- Fault-tolerant
- Secure
- Scalable

High Availability

"How many nines?"

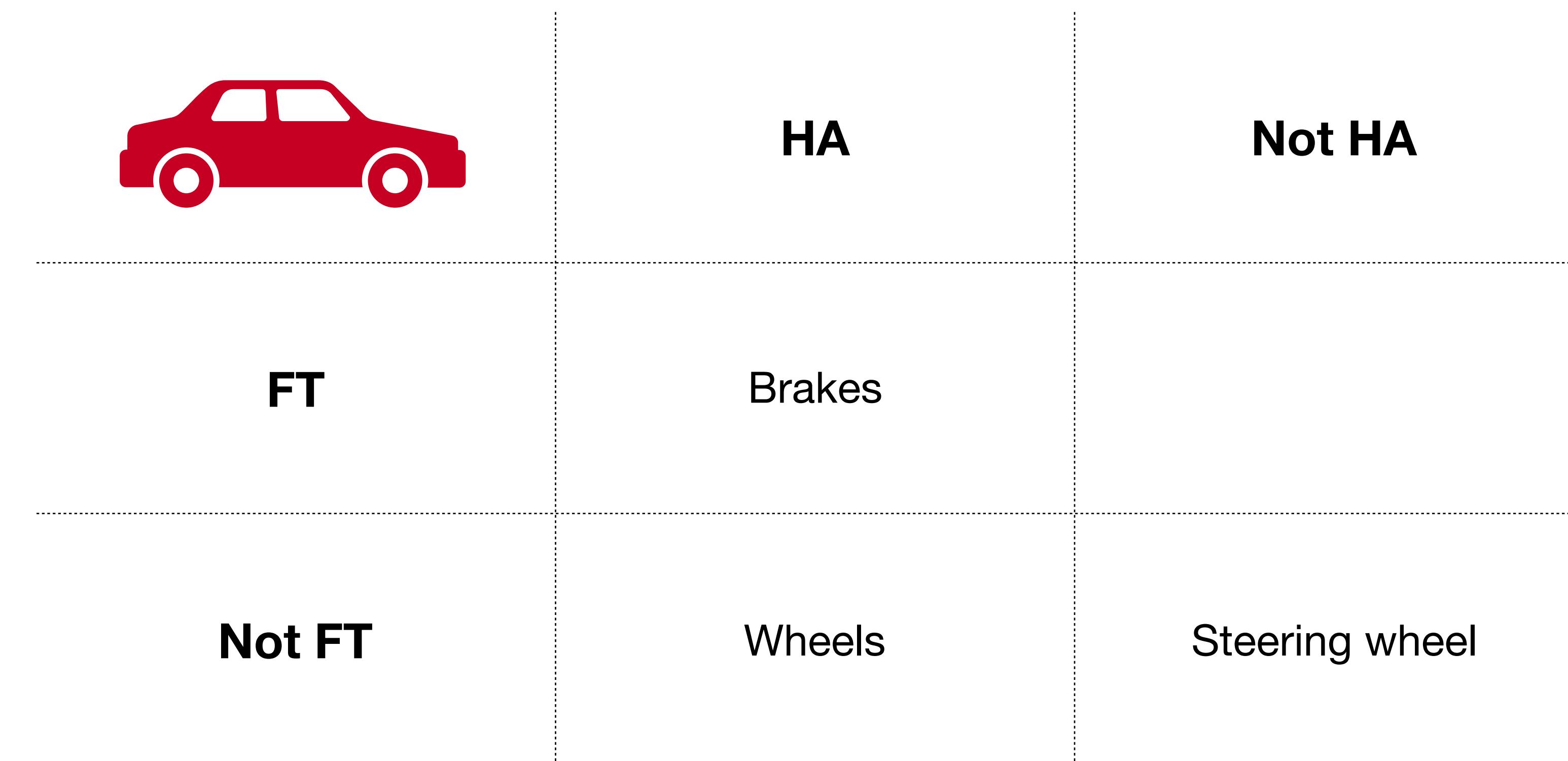
How many nines?

Uptime %	"Nines"	Downtime per day	Downtime per year
90%	1 nine	2.4 hours	36.5 days
99%	2 nines	14 minutes	3.65 days
99.9%	3 nines	86 seconds	8.76 hours
99.99%	4 nines	8.6 seconds	52.6 minutes
99.999%	5 nines	0.86 seconds	5.25 minutes
99.9999%	6 nines	8.6 milliseconds	31.5 seconds

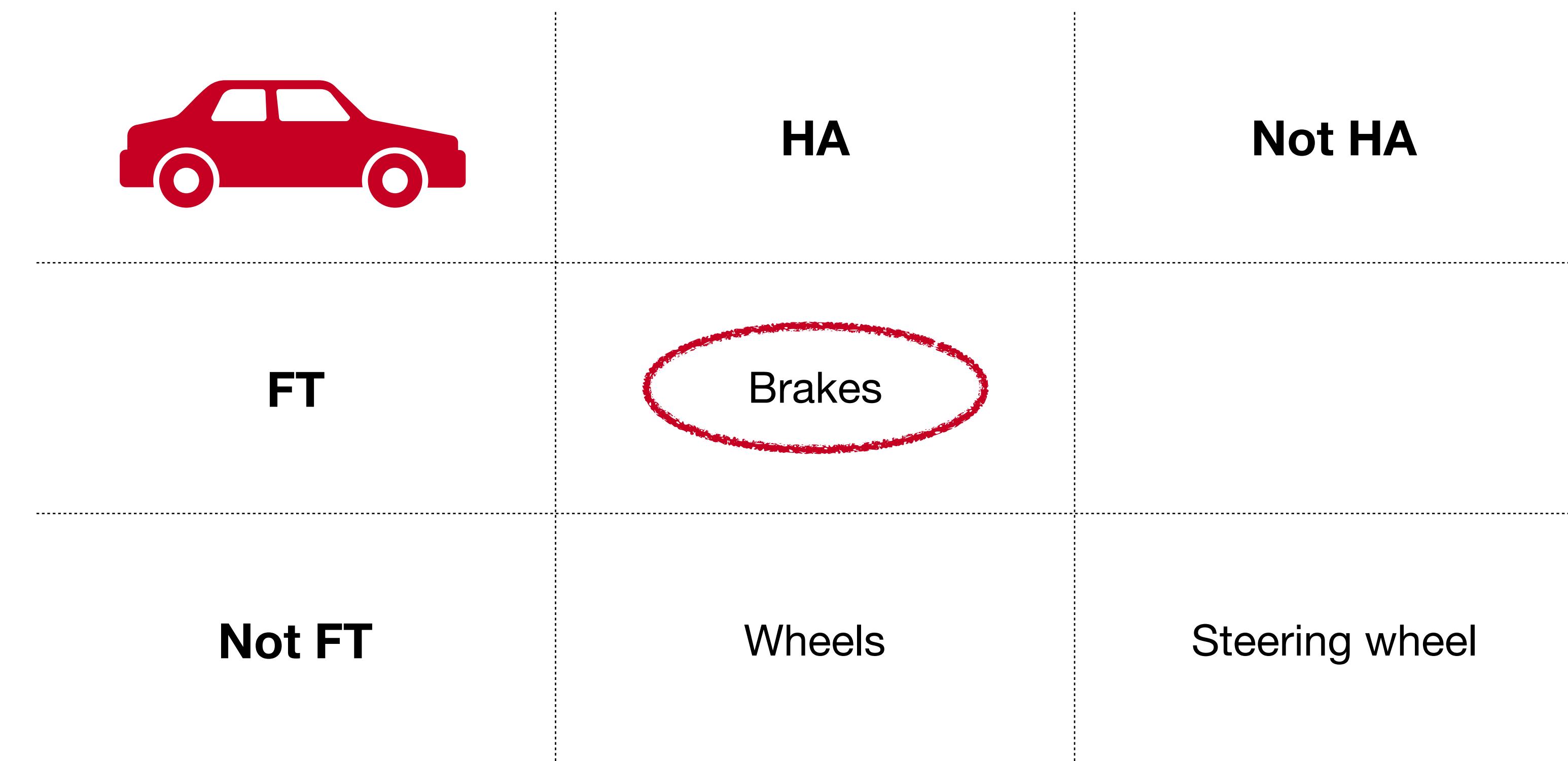
Fault Tolerance

A fault-tolerant system will continue to function properly despite failure of one or more of its components.

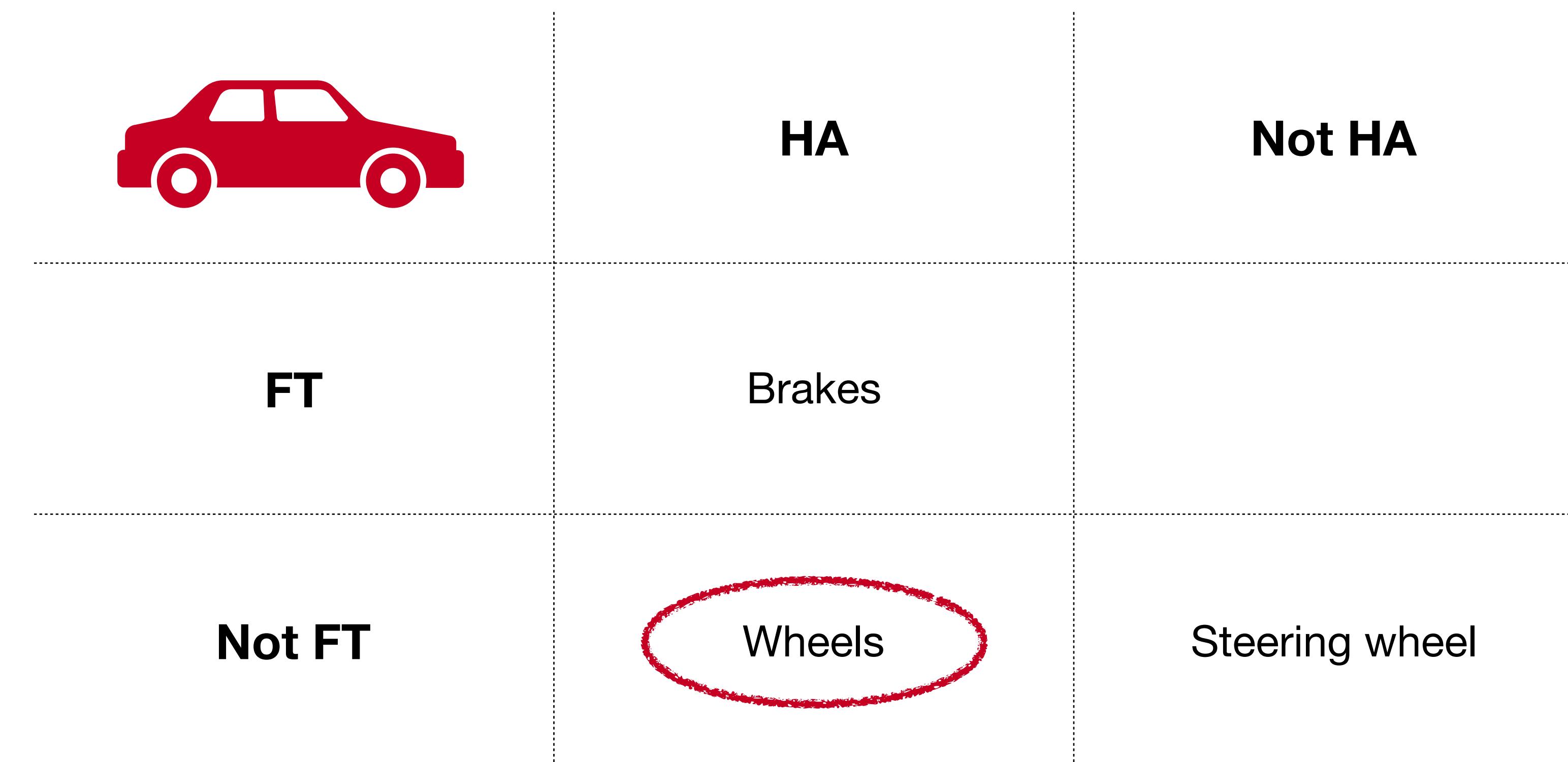
High Availability vs. Fault Tolerance



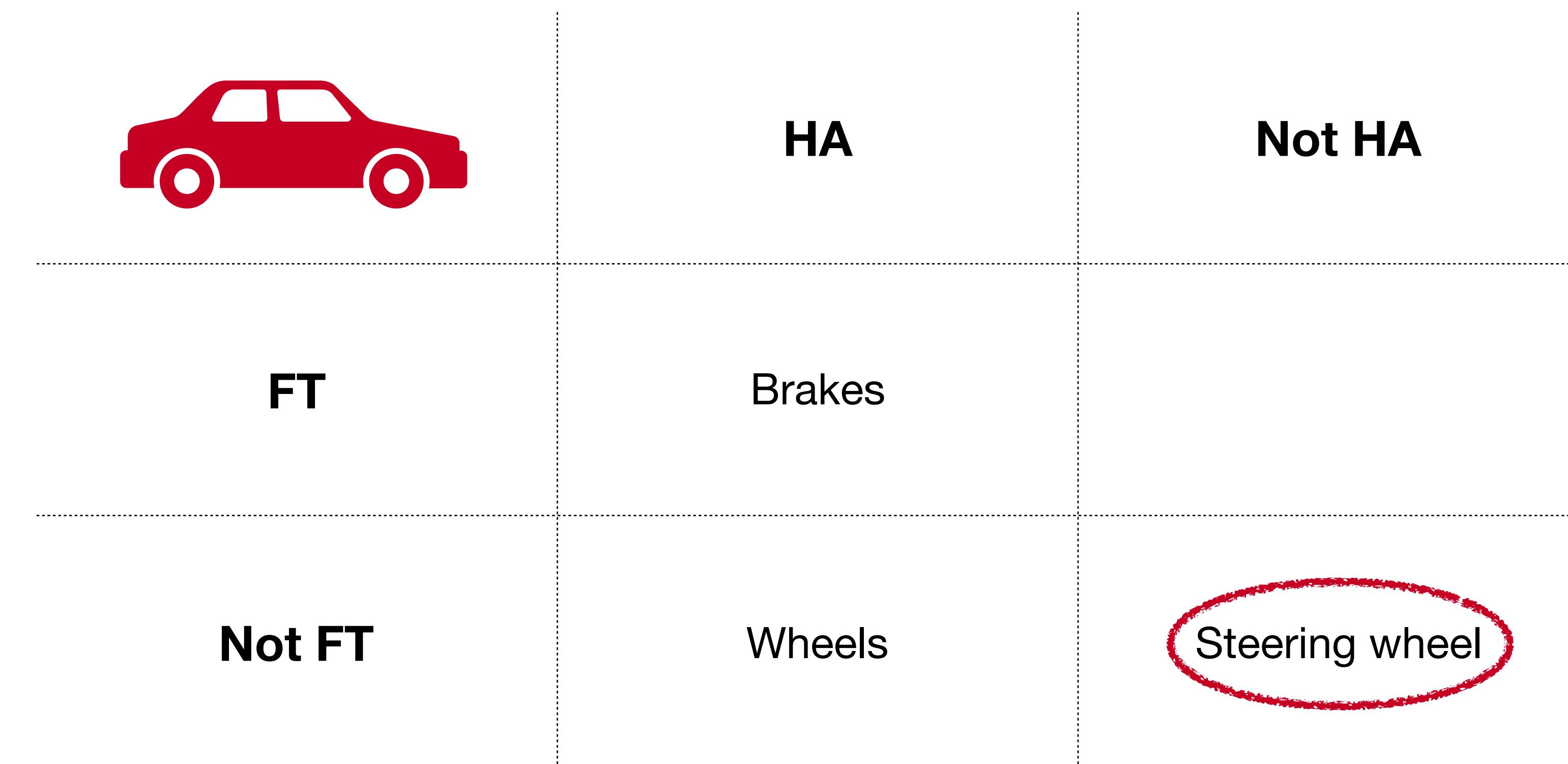
High Availability vs. Fault Tolerance



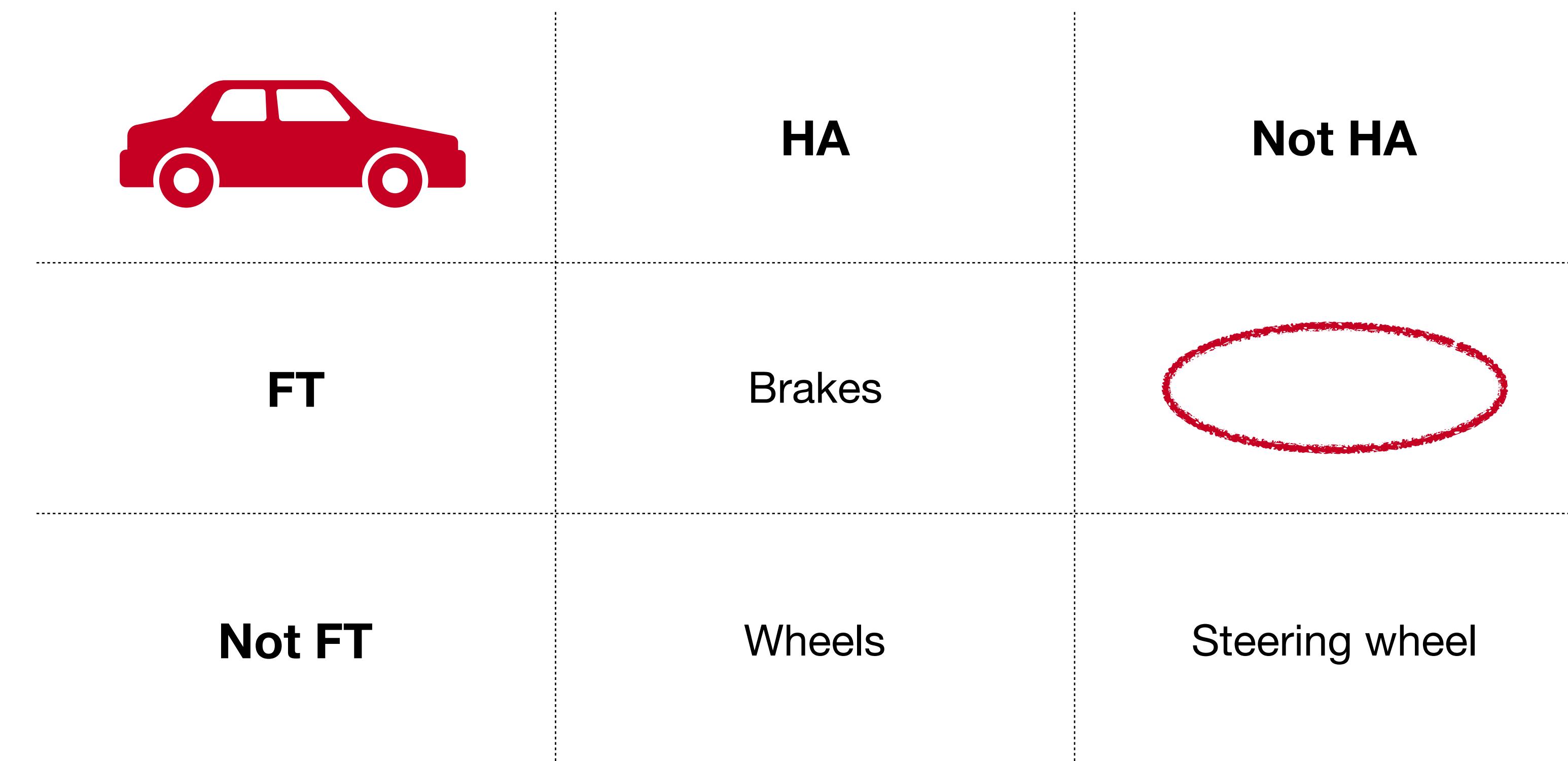
High Availability vs. Fault Tolerance



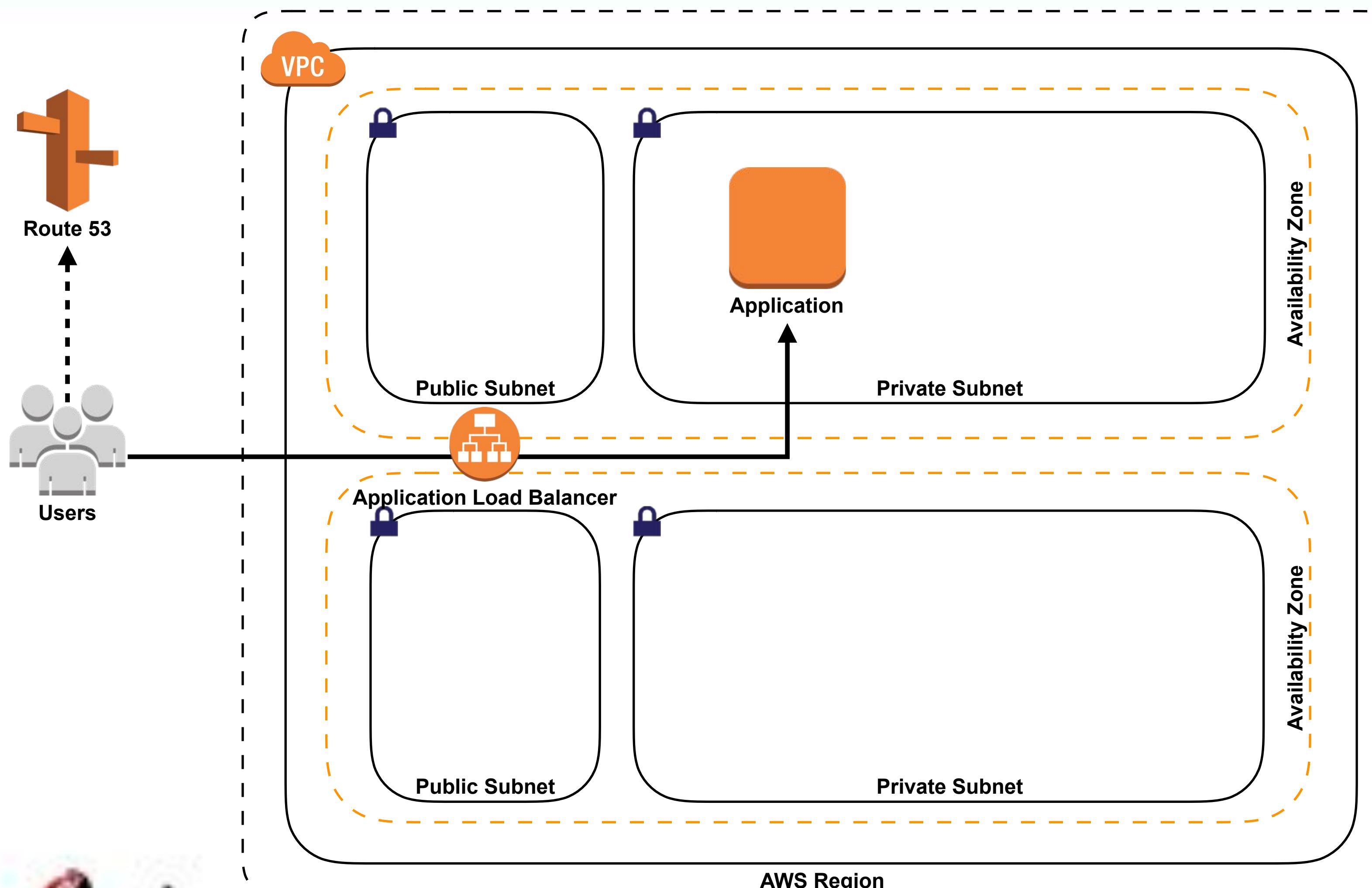
High Availability vs. Fault Tolerance



High Availability vs. Fault Tolerance

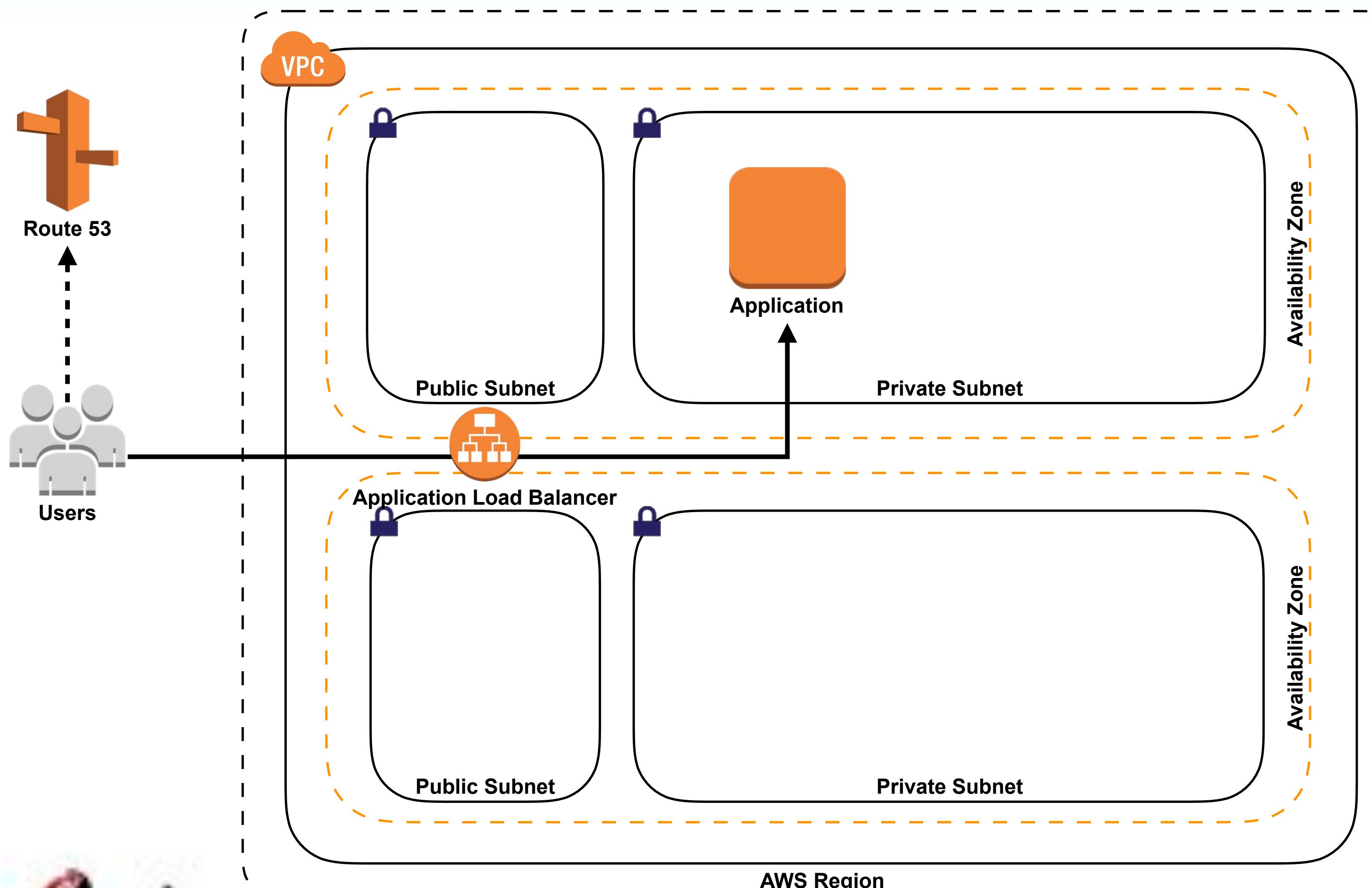


Checkpoint



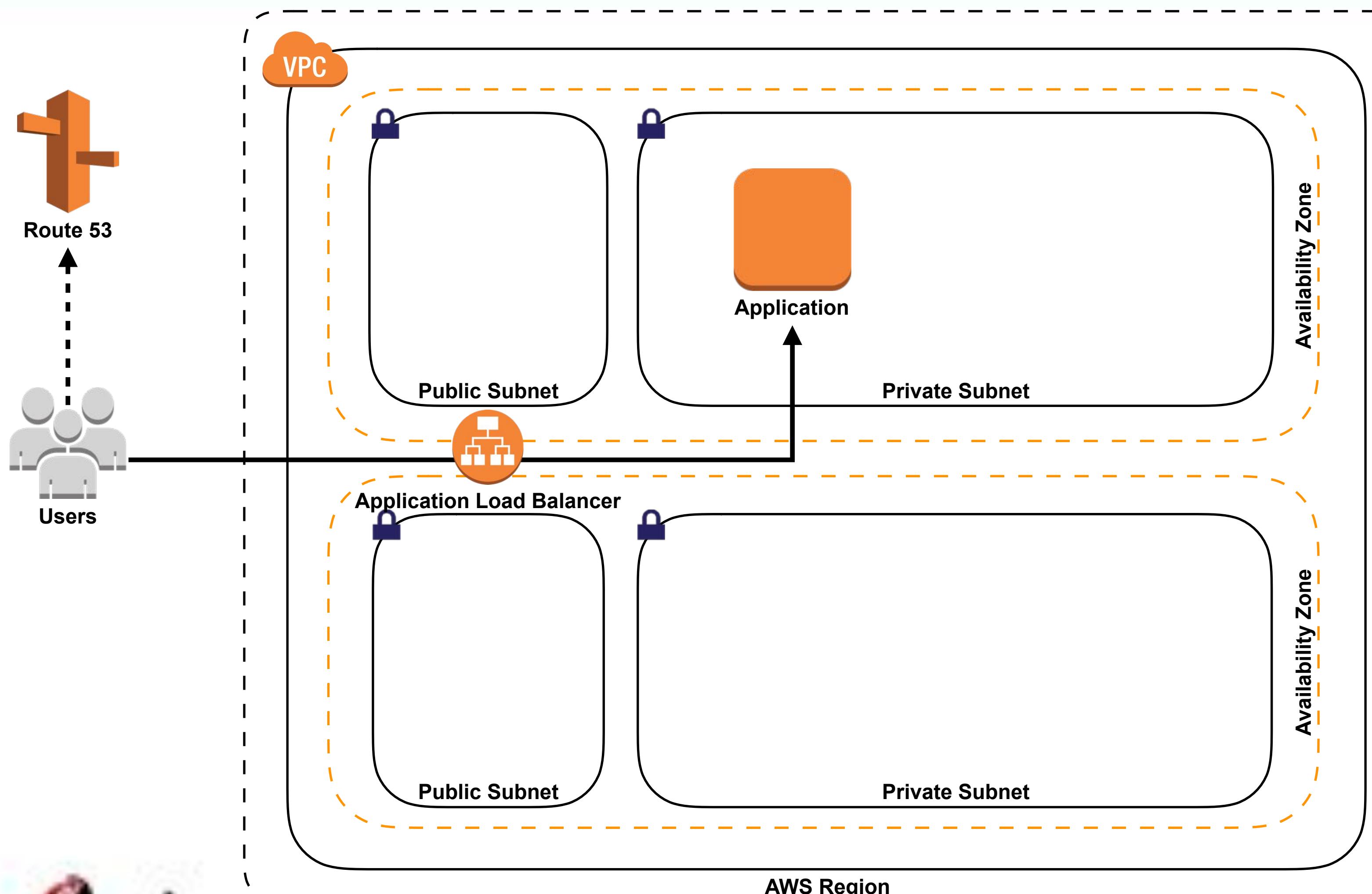
- Highly-available
- Fault-tolerant
- Secure
- Scalable

Checkpoint



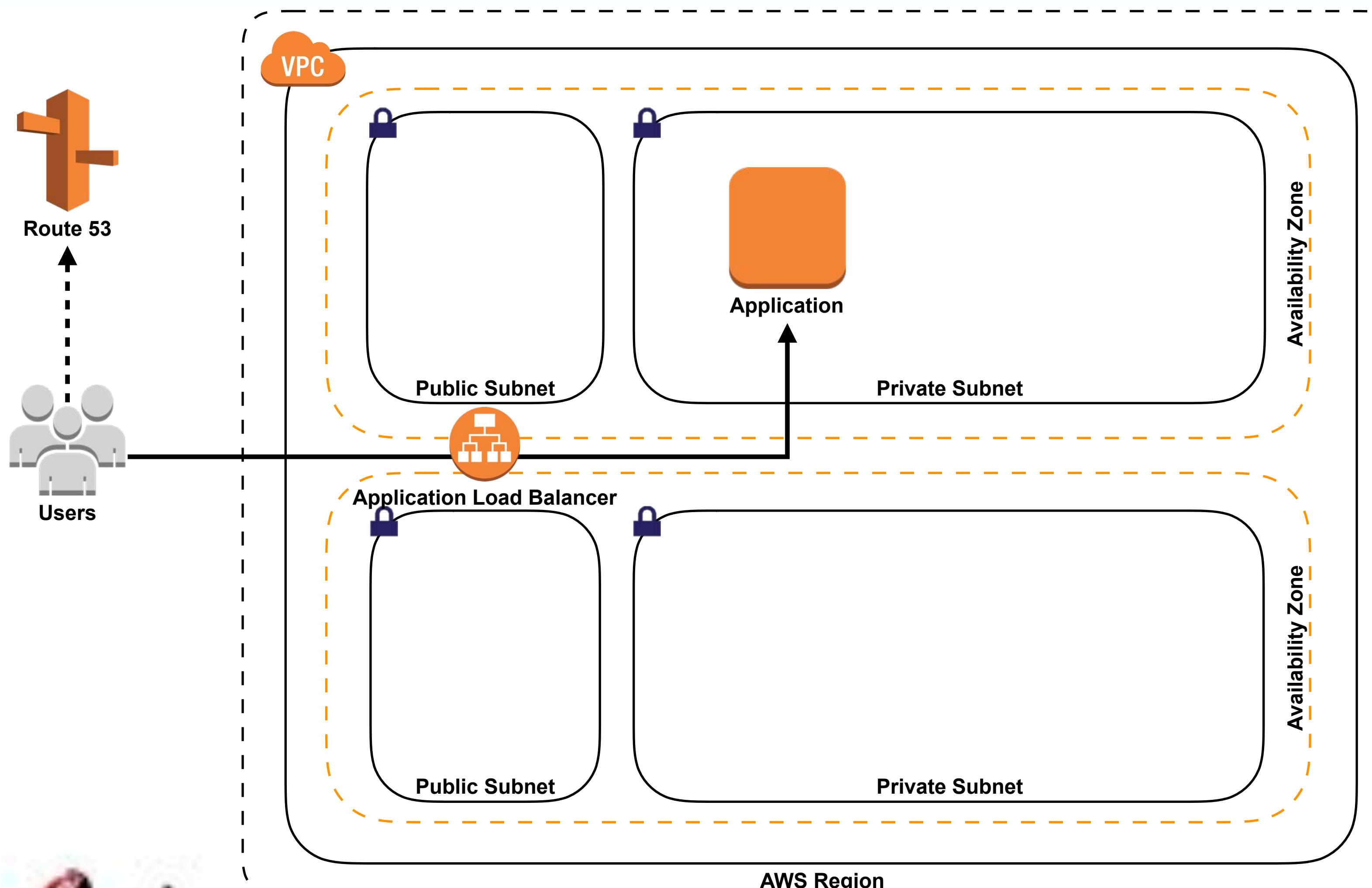
- Highly-available
- Fault-tolerant
- Secure
- Scalable

Checkpoint



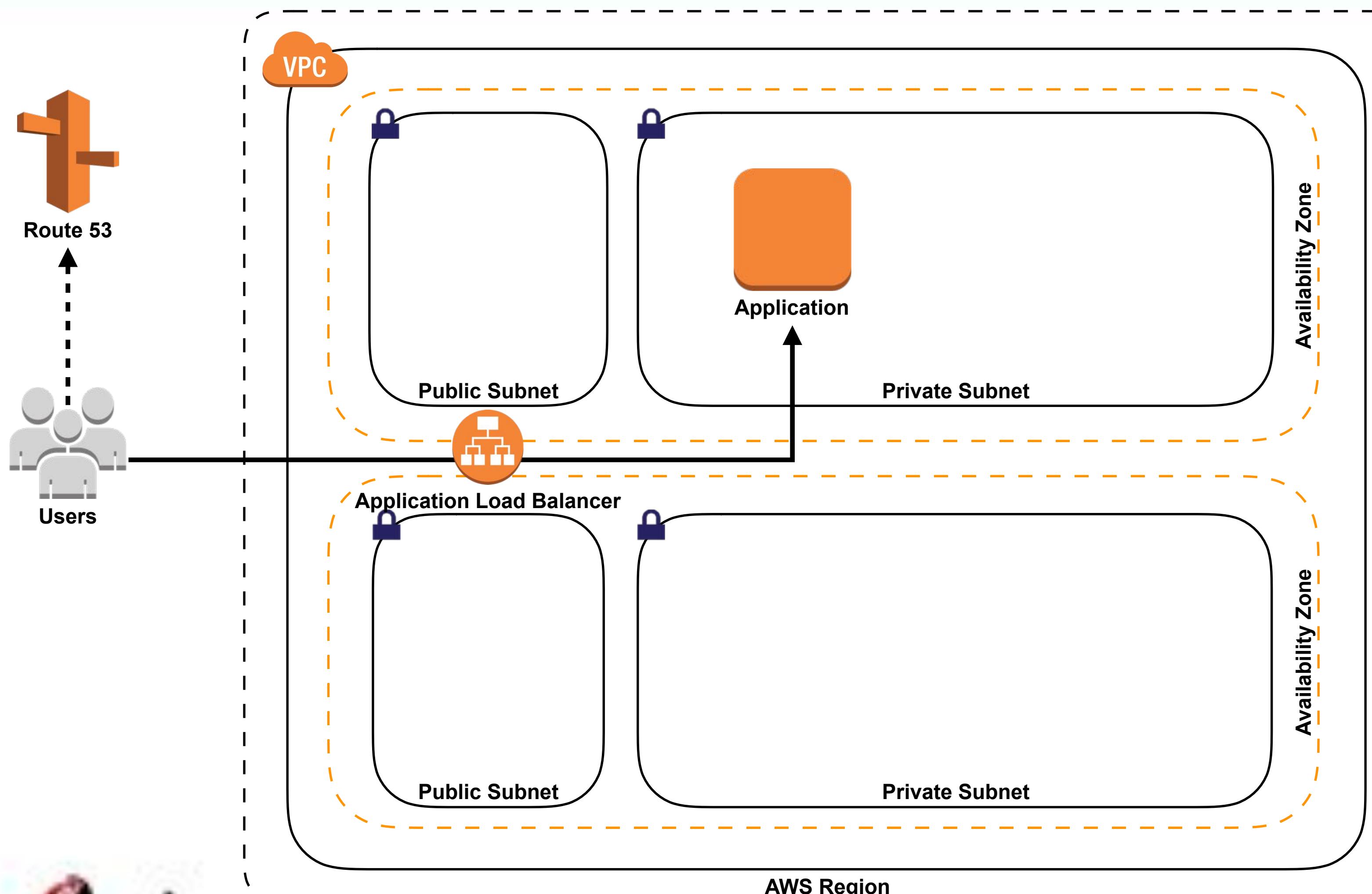
- Highly-available
- Fault-tolerant
- Secure
- Scalable

Checkpoint



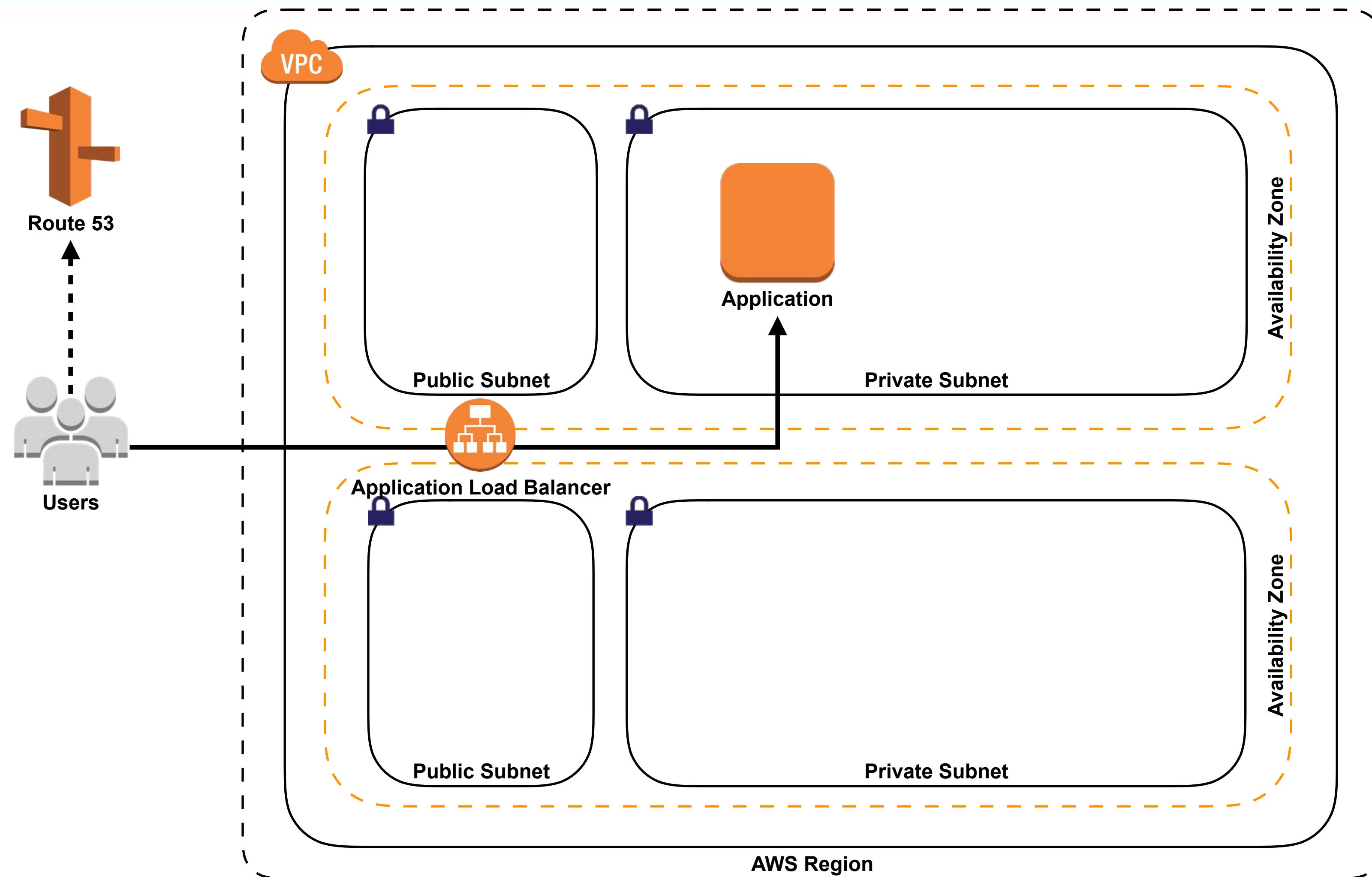
- Highly-available
- Fault-tolerant
- Secure
- Scalable

Checkpoint

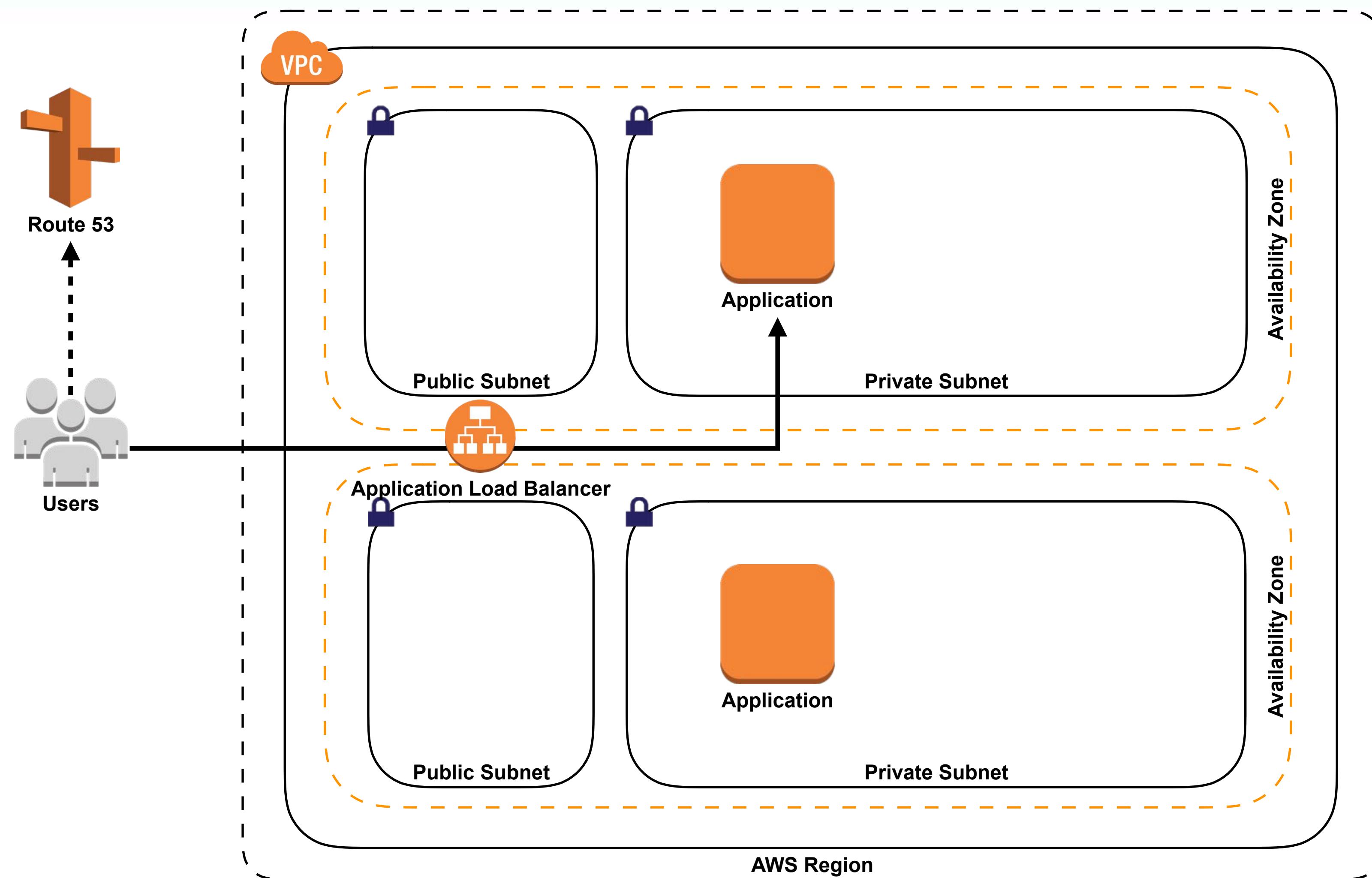


- Highly-available
- Fault-tolerant
- Secure
- Scalable

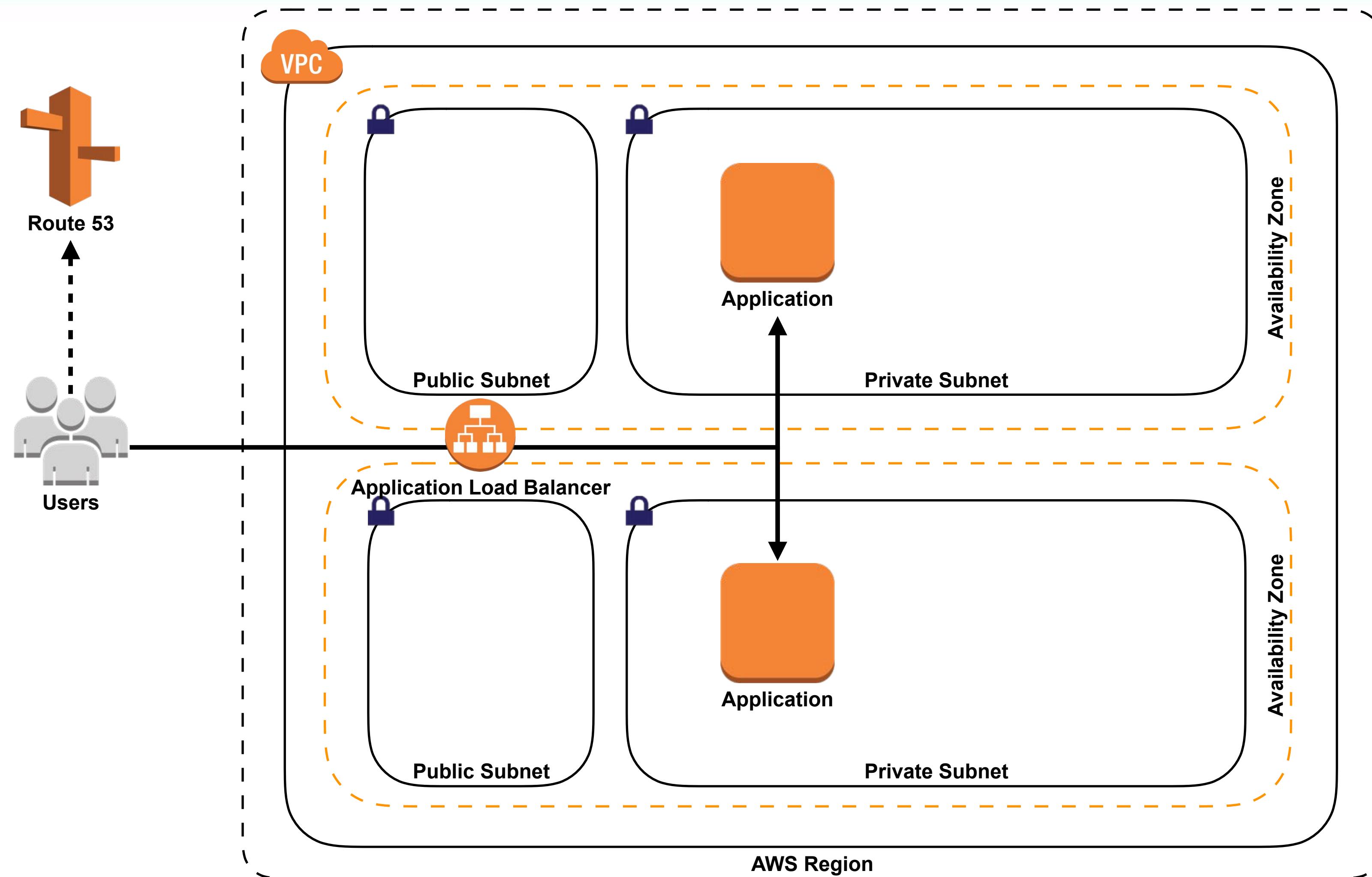
Checkpoint



Checkpoint



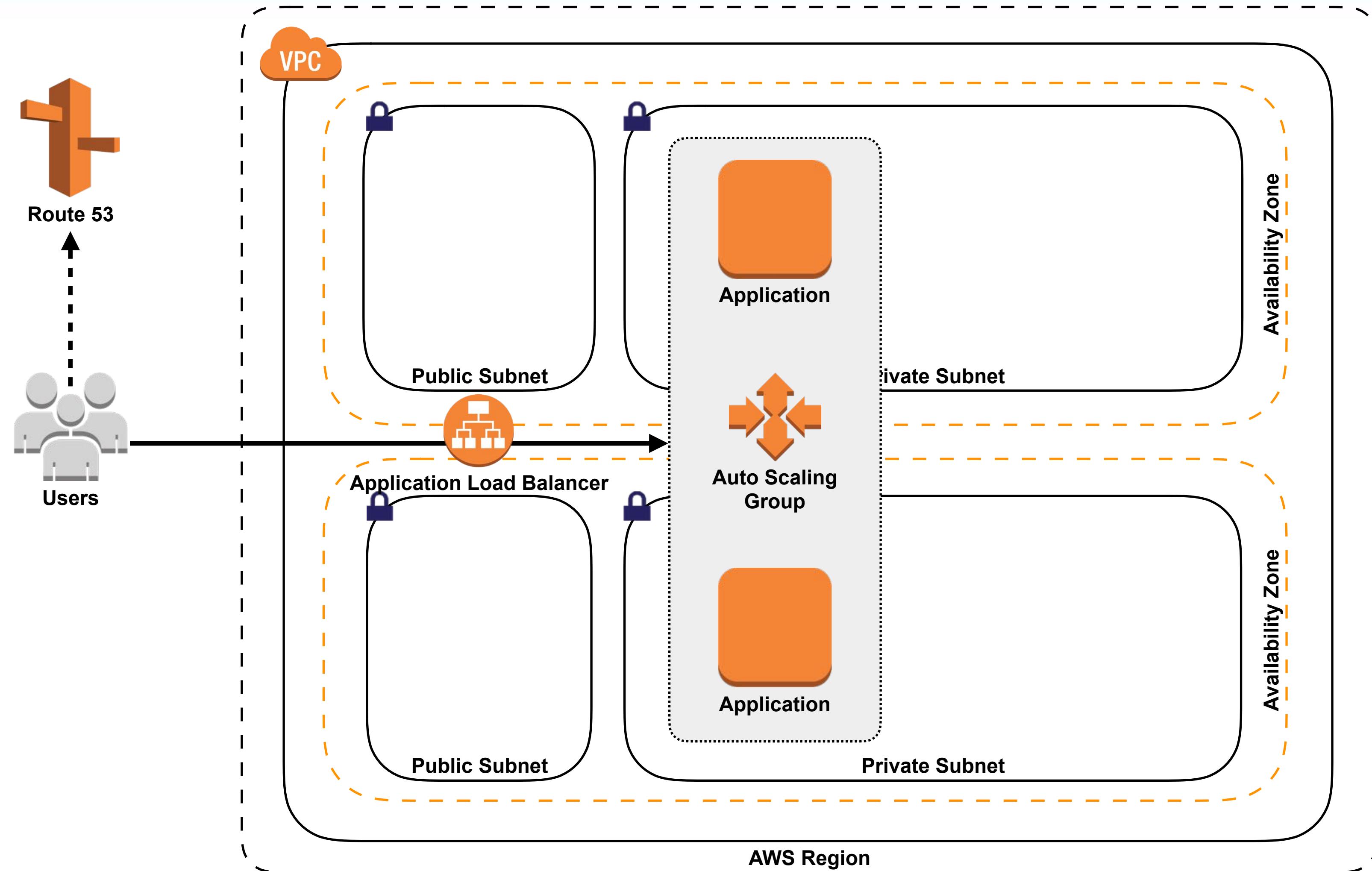
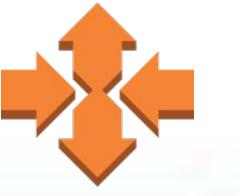
Checkpoint



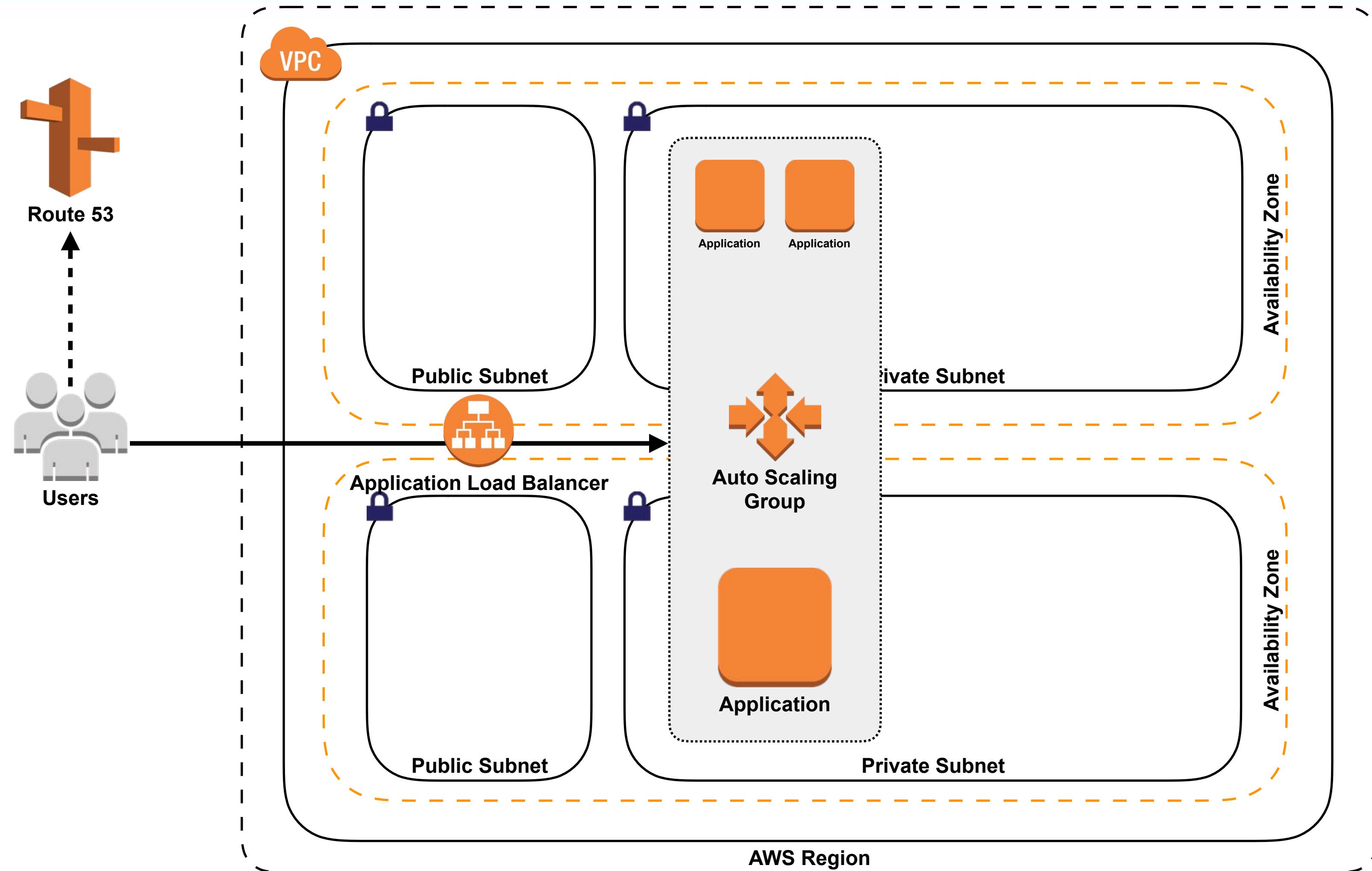
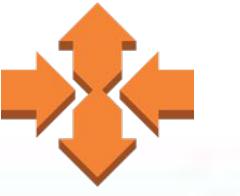
Auto Scaling Group (ASG)



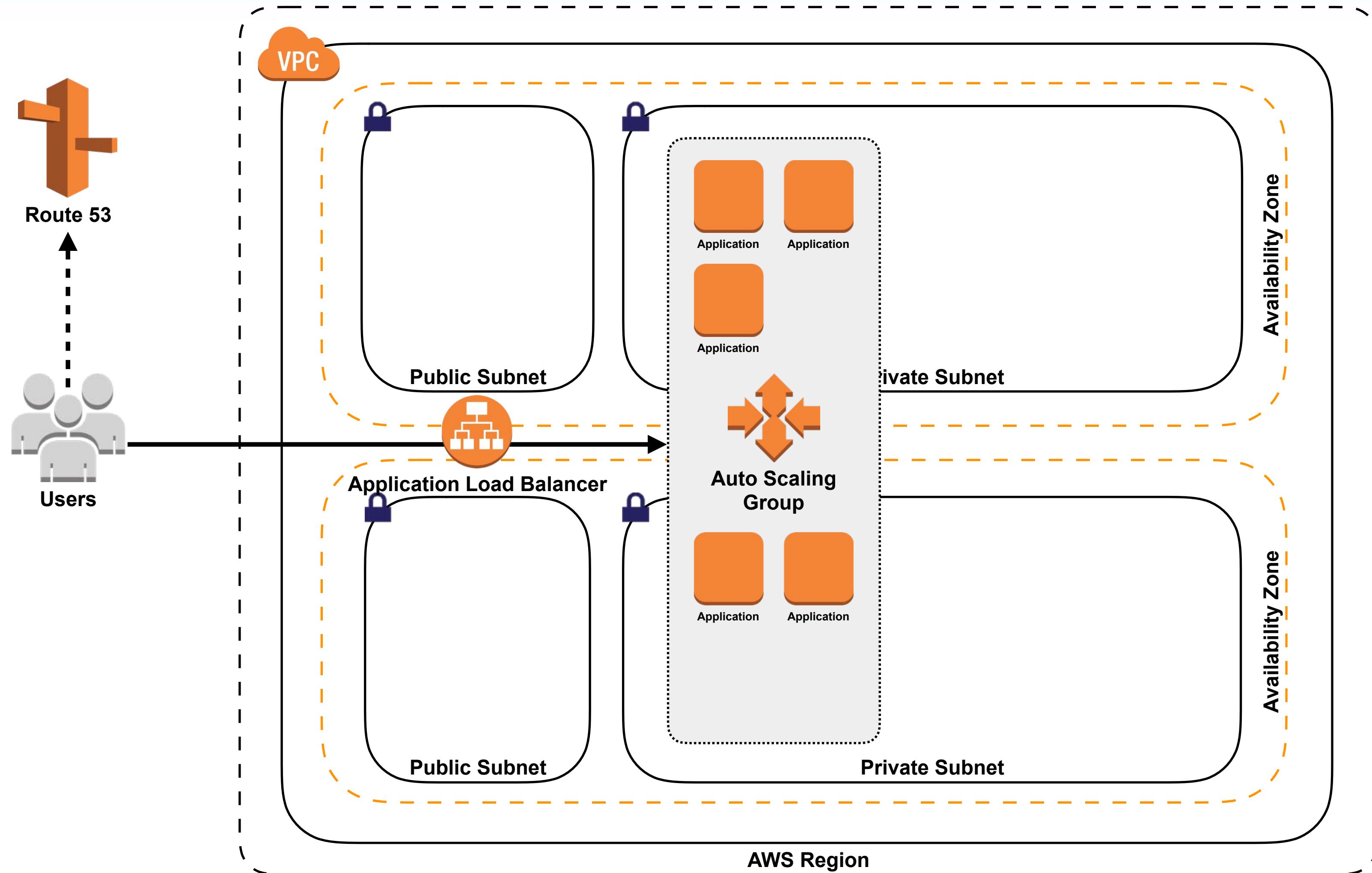
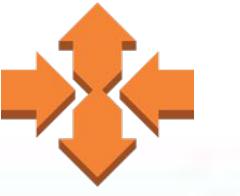
Auto Scaling Group (ASG)



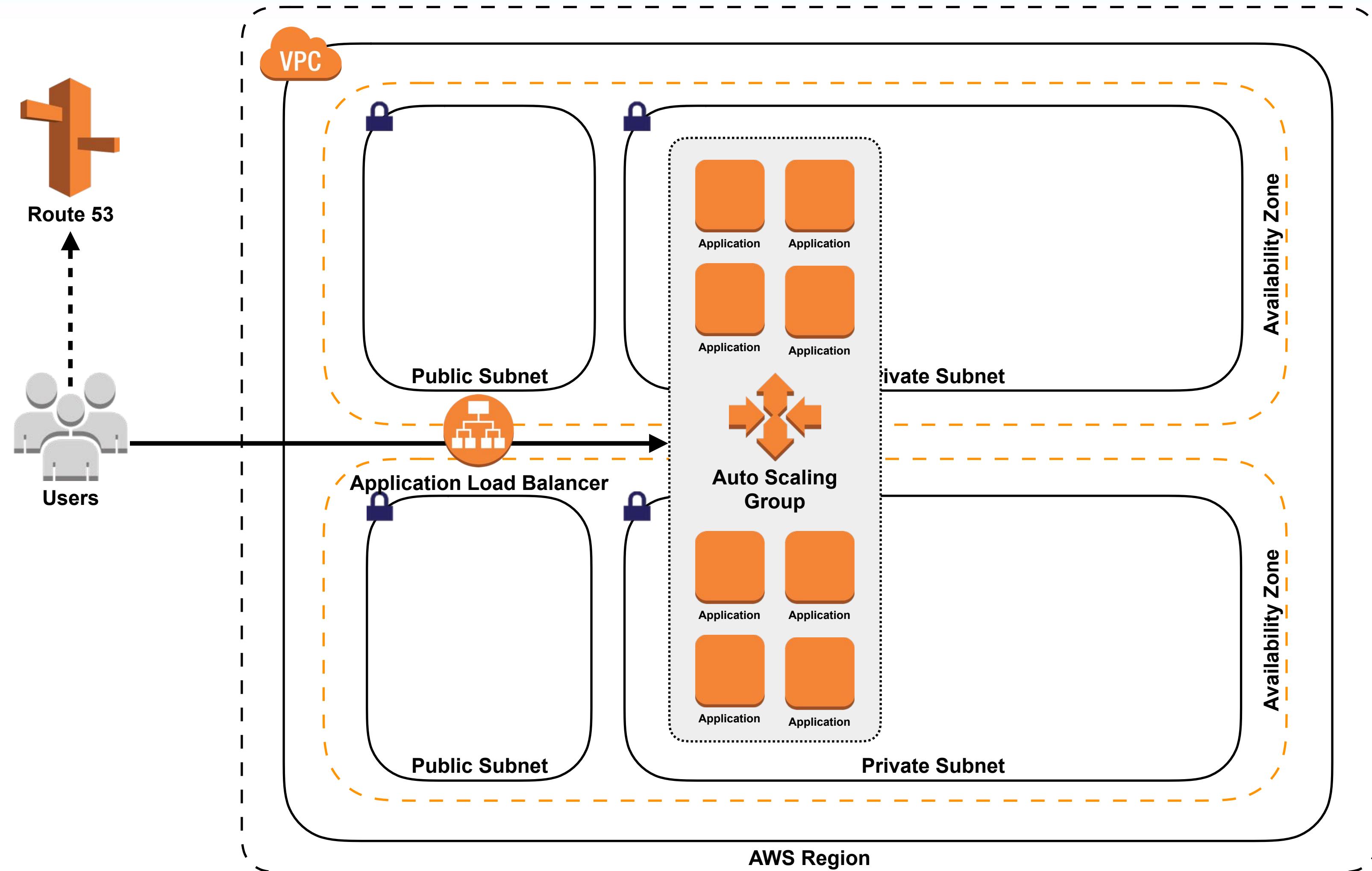
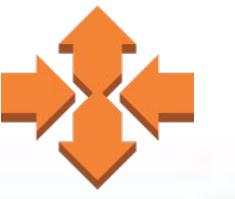
Auto Scaling Group (ASG)



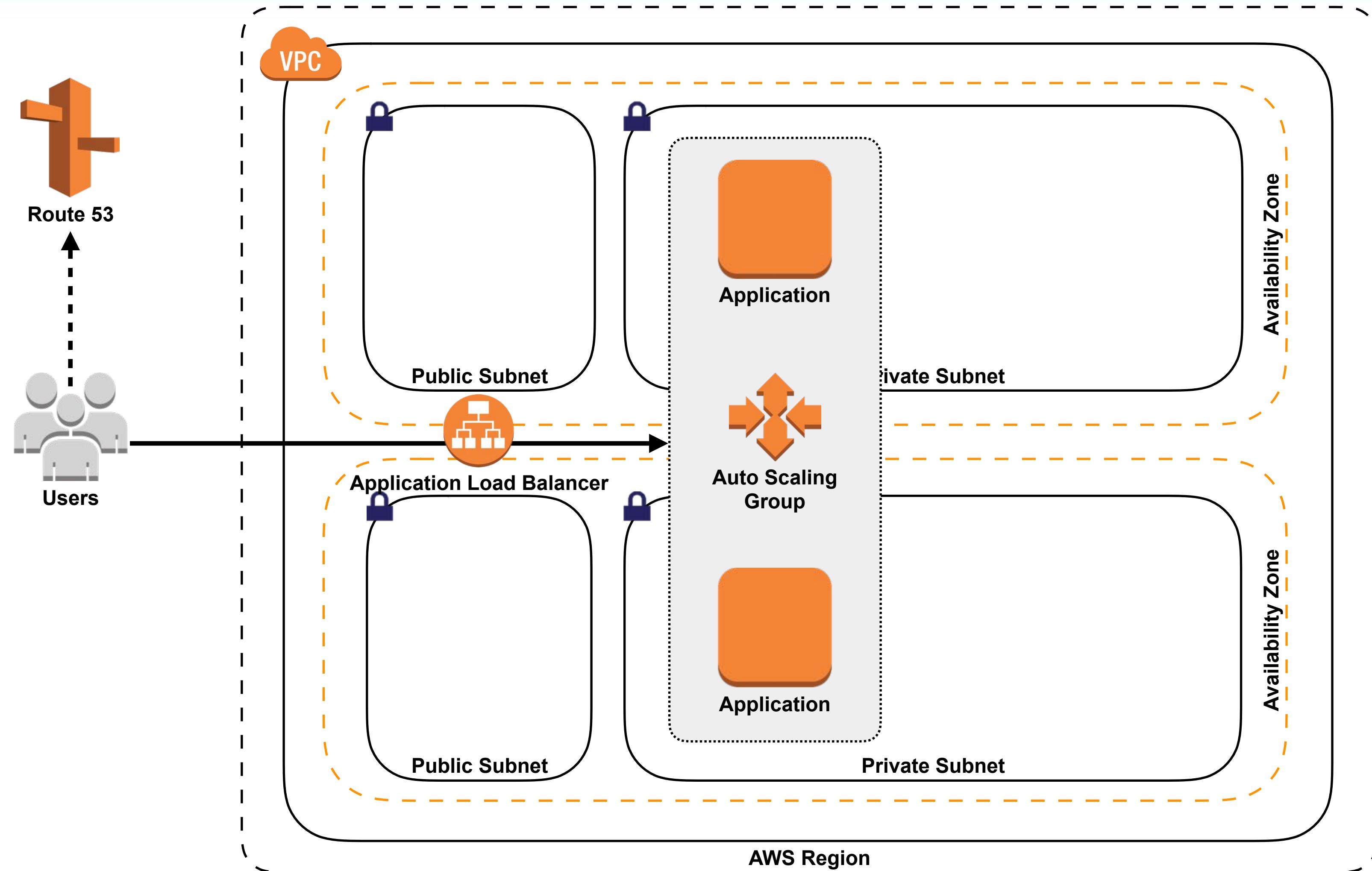
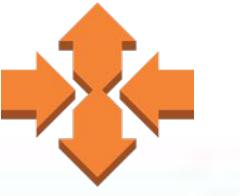
Auto Scaling Group (ASG)



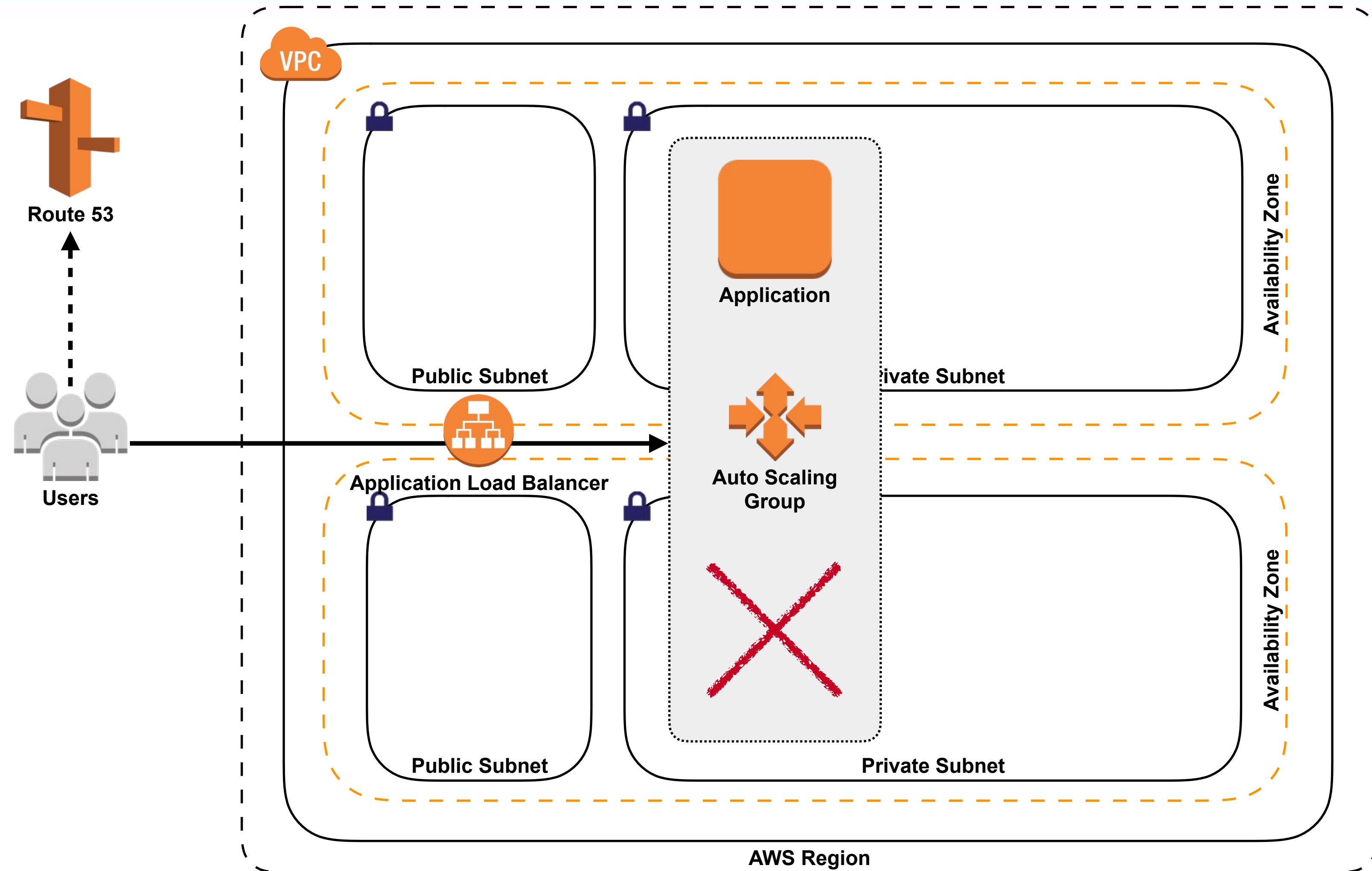
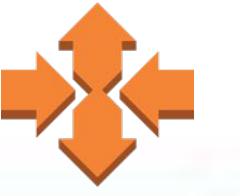
Auto Scaling Group (ASG)



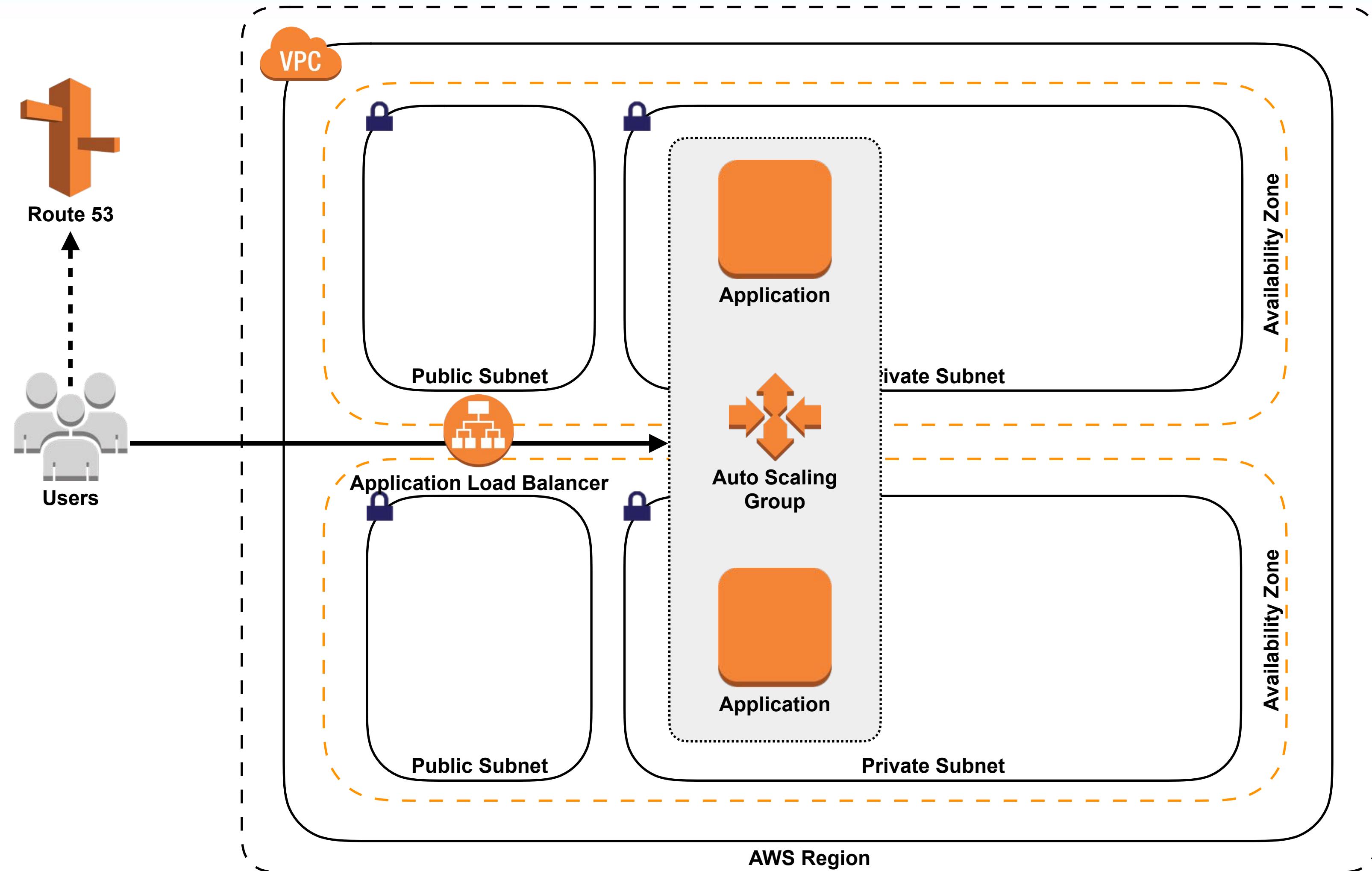
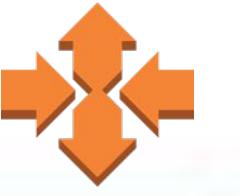
Auto Scaling Group (ASG)



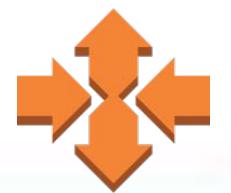
Auto Scaling Group (ASG)



Auto Scaling Group (ASG)



Auto Scaling Group (ASG) - Scale out



CPU-AddCapacity

Policy type: Step scaling

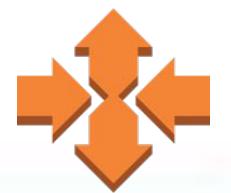
Execute policy when: AddCapacity-CPU-Utilization
breaches the alarm threshold: CPUUtilization ≥ 50 for 300 seconds
for the metric dimensions AutoScalingGroupName = myapp-asg

Take the action: Add 2 instances when $50 \leq \text{CPUUtilization} < 75$

Add 4 instances when $75 \leq \text{CPUUtilization} < +\infty$

Instances need: 300 seconds to warm up after each step

Auto Scaling Group (ASG) - Scale out



CPU-AddCapacity

Policy type: Step scaling

Execute policy when:

CloudWatch Metrics

CloudWatch Metrics

CloudWatch Metrics

CloudWatch Metrics

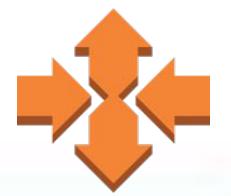
CPUUtilization >= 50 for 300 seconds

Take the action: Add 2 instances when $50 \leq \text{CPUUtilization} < 75$

Add 4 instances when $75 \leq \text{CPUUtilization} < +\infty$

Instances need: 300 seconds to warm up after each step

Auto Scaling Group (ASG) - Scale out



CPU-AddCapacity

Policy type: Step scaling

Execute policy when: AddCapacity-CPU-Utilization
breaches the alarm threshold: CPUUtilization ≥ 50 for 300 seconds
for the metric dimensions AutoScalingGroupName = myapp-asg

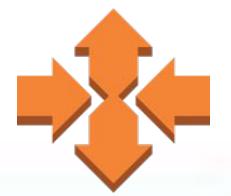
Take th

Add 2 instances when $50 \leq \text{CPUUtilization} < 75$

Add 4 instances when $75 \leq \text{CPUUtilization} < +\infty$

Instances need: 300 seconds to warm up after each step

Auto Scaling Group (ASG) - Scale in



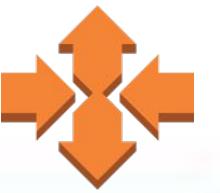
CPU-RemoveCapacity

Policy type: Step scaling

Execute policy when: RemoveCapacity-CPU-Utilization
breaches the alarm threshold: CPUUtilization < 75 for 300 seconds
for the metric dimensions AutoScalingGroupName = myapp-asg

Take the action: Remove 4 instances when $75 \geq \text{CPUUtilization} > 50$
Remove 2 instances when $50 \geq \text{CPUUtilization} > -\infty$

Auto Scaling Group (ASG) - Scale in



CPU-RemoveCapacity

Execute policy when:

ng

apacity-C

breaches the alarm

CPUUtilization < 75 for 300 seconds

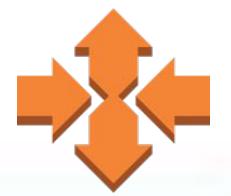
for the metric dimensions AutoScalingGroupName = myapp-asg

Take the action:

Remove 4 instances when $75 \geq \text{CPUUtilization} > 50$

Remove 2 instances when $50 \geq \text{CPUUtilization} > -\infty$

Auto Scaling Group (ASG) - Scale in



CPU-RemoveCapacity

Policy type: Step scaling

Execute policy when: RemoveCapacity-CPU-Utilization
breaches the alarm threshold: CPUUtilization < 75 for 300 seconds
for the metric dimensions AutoScalingGroupName = myapp-asg

Take the ac

Remove 4 instances when $75 \geq \text{CPUUtilization} > 50$
Remove 2 instances when $50 \geq \text{CPUUtilization} > -\infty$

Amazon CloudWatch



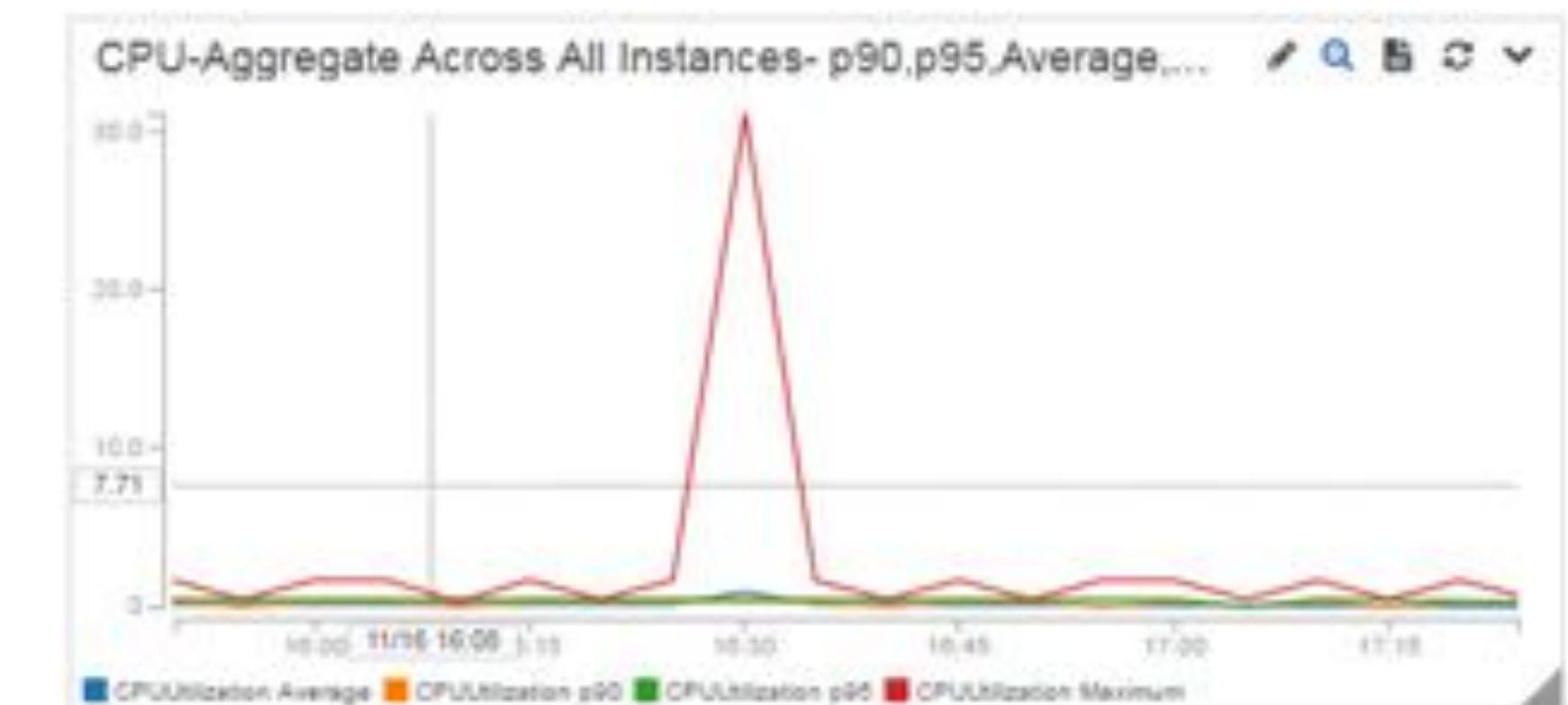
Amazon CloudWatch

- Monitoring service



Amazon CloudWatch

- Monitoring service
- Collect and display metrics



Amazon CloudWatch

- Monitoring service
- Collect and display metrics
- Alarms

Create Alarm

1. Select Metric 2. Define Alarm

Alarm Threshold

Provide the details and threshold for your alarm. Use the graph on the right to help set the appropriate threshold.

Name: my-app-cpu

Description: Alarm when CPU goes above 75%

Whenever: CPUUtilization
is: >= 75
for: 1 out of 1 datapoints

Additional settings

Provide additional configuration for your alarm.

Treat missing data as: missing

Actions

Define what actions are taken when your alarm changes state.

Notification

Whenever this alarm: State is ALARM
Send notification to: myapp-devops-alert

+ Notification + AutoScaling Action + EC2 Action

Amazon CloudWatch

- Monitoring service
- Collect and display metrics
- Alarms

Create Alarm

1. Select Metric 2. Define Alarm

Whenever: CPUUtilization
is: \geq 75
for: 1 out of datapoints [i](#)

Additional settings

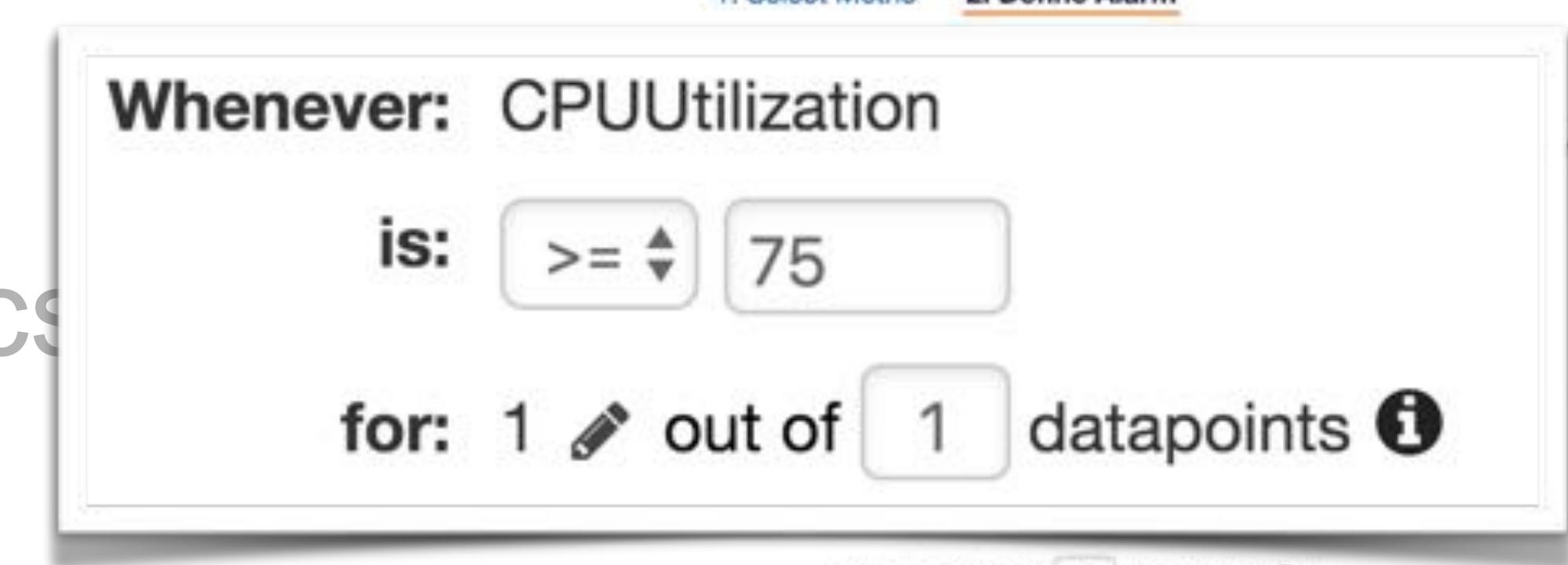
Provide additional configuration for your alarm.

Notification [Delete](#)

Whenever this alarm: State is ALARM

Send notification to: myapp-devops-alert [New list](#) [Enter list](#) [i](#)

[+ Notification](#) [+ AutoScaling Action](#) [+ EC2 Action](#)



Amazon CloudWatch

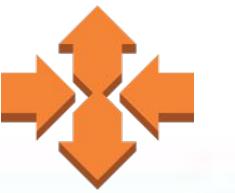
- Monitoring service
- Collect and display metrics
- Alarms
- Gather and store server logs

with awslogs agent

The screenshot shows the CloudWatch Logs interface. On the left, a sidebar menu includes Dashboard, Alarms (with ALARM and INSUFFICIENT status), Billing, Metrics, and Logs (which is selected and highlighted in orange). Below the sidebar, a breadcrumb navigation path reads Log Groups > Streams for /var/log/messages > Events for i-f9cd4912. A date/time selector shows 2014/09/16 17:15:51 UTC (GMT). The main content area displays a table of log events. The columns are Creation Time and Event Data. The table lists 20 entries, each showing a timestamp from 2014-09-16 17:15:51 UTC and a corresponding log message. The messages are mostly kernel logs related to system initialization and memory management.

Creation Time	Event Data
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: imklog 5.8.10, log source
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 rsyslogd: [origin software="rsyslogd"]
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] Initializing c
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] Initializing c
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] Command line:
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] BIOS-pro
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] BIOS-e820: [mem
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] BIOS-e820: [mem
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] BIOS-e820: [mem
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] BIOS-e820: [mem
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] BIOS-e820: [mem
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] NX (Execute Bi
2014-09-16 17:15:51 UTC	* Sep 16 00:14:16 ip=10-144-226-204 kernel: [0.000000] SMBIOS 2.4 pre

Auto Scaling Group (ASG) - Target tracking



Maintain-CPU-at-40

Policy type: Target Tracking scaling

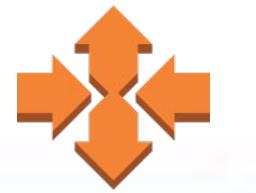
Execute policy when: As required to maintain Average CPU Utilization at 40

Take the action: Add or remove instances as required

Instances need: 300 seconds to warm up after scaling

Disable scale-in: No

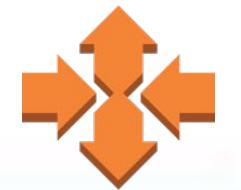
Auto Scaling Group (ASG) - Target tracking



Maintain-CPU-at-40

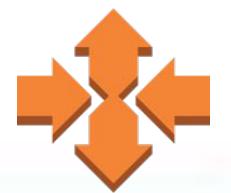
Policy type:	Target Tracking scaling
Execute policy when:	Average CPU Utilization at 40
Take the action:	Add or remove instances as required
Instances need:	300 seconds to warm up after scaling
Disable scale-in:	No

Auto Scaling Group (ASG) - Scheduling



Name	WeekendTraffic		
Auto Scaling Group	myapp-asg		
Provide at least one of Min, Max and Desired Capacity			
Min	8		
Max	16		
Desired Capacity			
Recurrence	Every week ▾ (Cron) 0 12 * * Fri		
Start Time	2018-02-09	12 : 00	UTC Specify the start time in UTC
	The first time this scheduled action will run		
End Time	2018-02-12	06 : 00	UTC Specify the end time in UTC Cancel

Auto Scaling Group (ASG) - Scheduling



Name

Auto Scaling Group

Provide at least one of Min. Max and Desired Capacity

Min

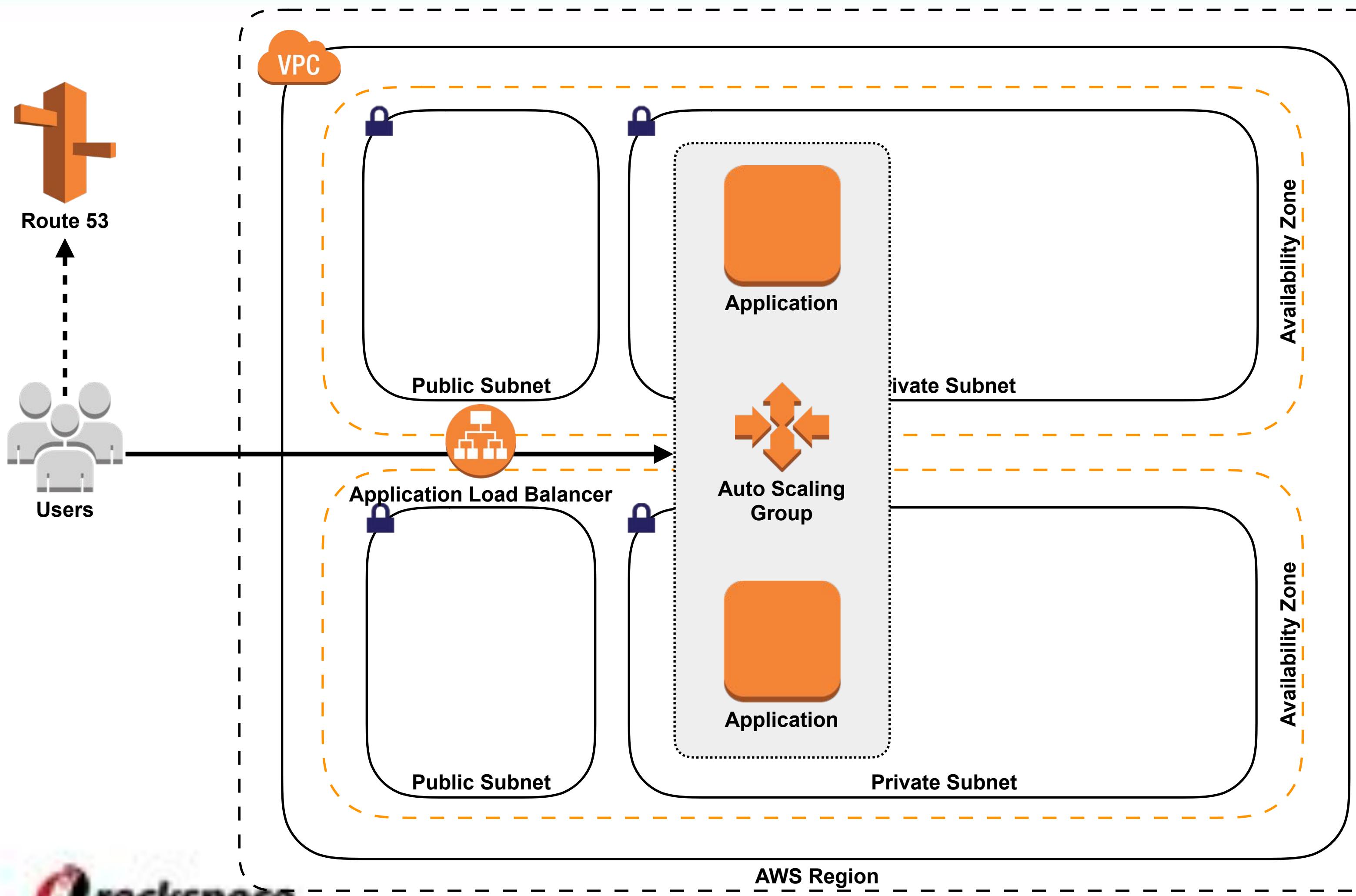
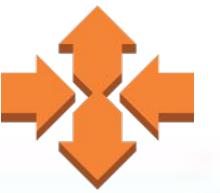
Recurrence (Cron) 0 12 * * Fri

Start Time UTC The first time this scheduled action

End Time UTC Time in UTC

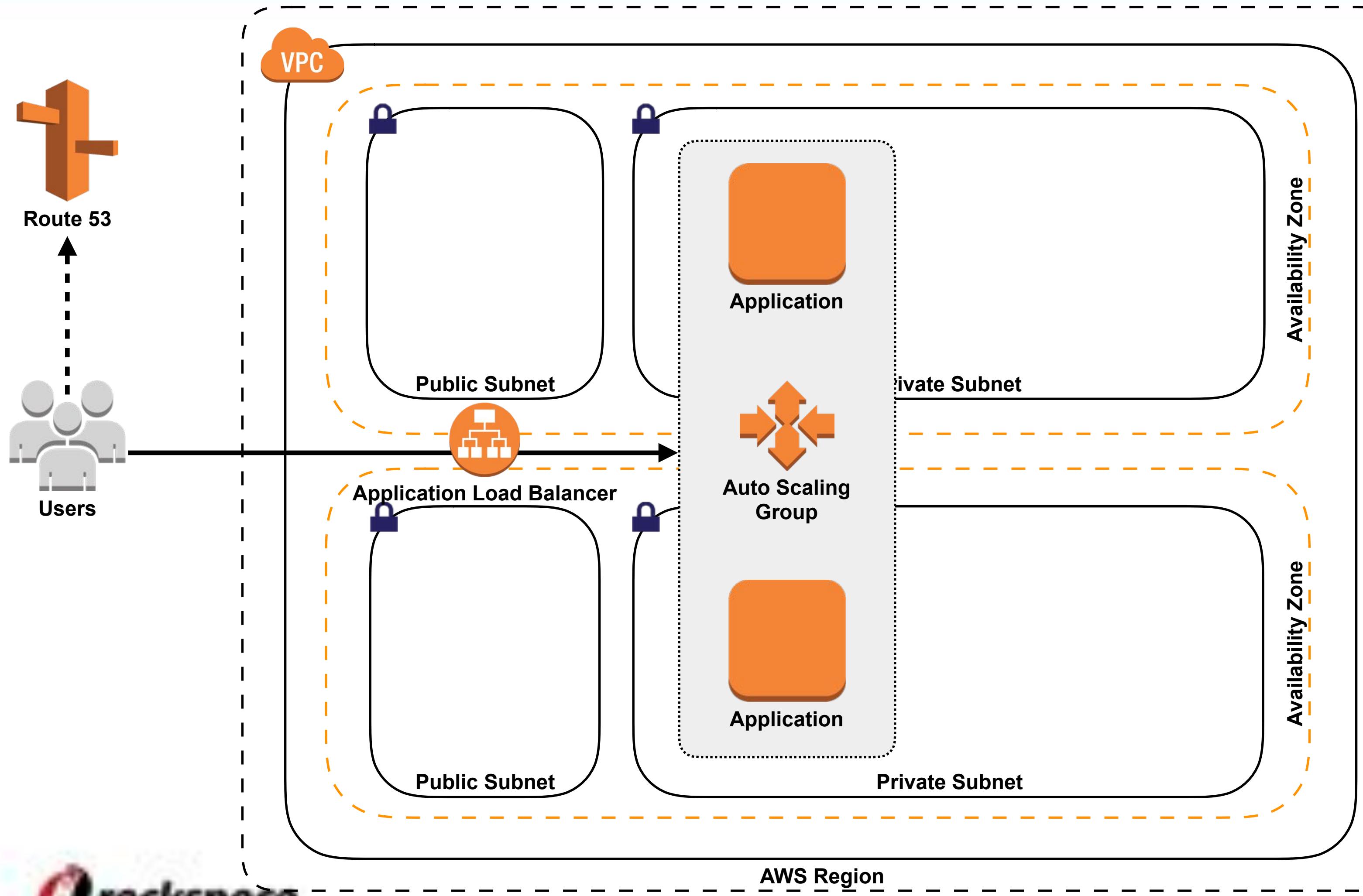
Time in UTC [Cancel](#)

Auto Scaling Group (ASG)



- Highly-available
- Fault-tolerant
- Secure
- Scalable

Auto Scaling Group (ASG)



- Highly-available
- Fault-tolerant
- Secure
- Scalable

Relational Database Service (RDS)



Relational Database Service (RDS)

- RDBMS as a service



Relational Database Service (RDS)

- RDBMS as a service
- Automated setup and patching

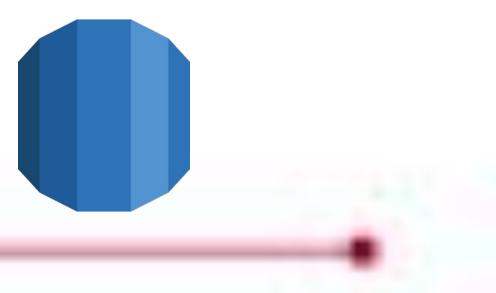


Relational Database Service (RDS)

- RDBMS as a service
- Automated setup and patching
- Managed backups



Managed backups



RDS > Snapshots

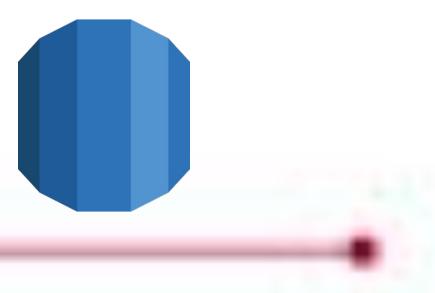
Snapshots (1)

Owned by Me Snapshot Actions Create snapshot

Q Filter snapshots < 1 > ⌂ C

<input type="checkbox"/>	Snapshot	DB Instance or Cluster	Snapshot Creation Time	Status	Type	Engine
<input type="checkbox"/>	rds:mytestdb-2017-12-31-00-37	mytestdb	Mon Dec 31 19:38:55 GMT-500 2017	available	automated	MySQL
<input type="checkbox"/>	rds:mytestdb-2018-01-01-00-37	mytestdb	Tue Jan 01 19:38:55 GMT-500 2018	available	automated	MySQL
<input type="checkbox"/>	rds:mytestdb-2018-01-02-00-37	mytestdb	Wed Jan 02 19:38:55 GMT-500 2018	available	automated	MySQL
<input type="checkbox"/>	rds:mytestdb-2018-01-03-00-37	mytestdb	Thu Jan 03 19:38:55 GMT-500 2018	available	automated	MySQL
<input type="checkbox"/>	rds:mytestdb-2018-01-04-00-37	mytestdb	Fri Jan 04 19:38:55 GMT-500 2018	available	automated	MySQL
<input type="checkbox"/>	rds:mytestdb-2018-01-05-00-37	mytestdb	Sat Jan 05 19:38:55 GMT-500 2018	available	automated	MySQL
<input type="checkbox"/>	rds:mytestdb-2018-01-06-00-37	mytestdb	Sun Jan 06 19:38:55 GMT-500 2018	available	automated	MySQL

Managed backups



The screenshot shows the AWS RDS Snapshots interface. On the left, there's a sidebar with 'RDS' and 'Snapshots'. The main area has a heading 'Snapshots (1)' and a search bar 'Filter snapshots'. Below is a table with one row:

	Snapshot	DB Instance
<input type="checkbox"/>	rds:mytestdb-2017-12-31-00-37	mytestdb

A modal window titled 'Snapshot Creation Time' is open over the table, listing the creation times for each snapshot:

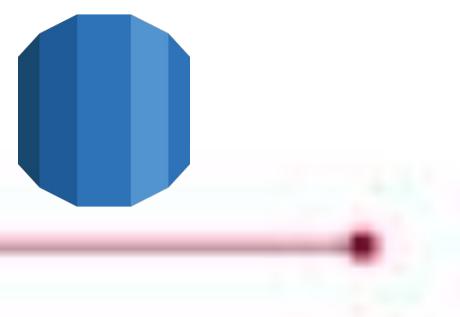
- Mon Dec 31 19:38:55 GMT-500 2017
- Tue Jan 01 19:38:55 GMT-500 2018
- Wed Jan 02 19:38:55 GMT-500 2018
- Thu Jan 03 19:38:55 GMT-500 2018
- Fri Jan 04 19:38:55 GMT-500 2018
- Sat Jan 05 19:38:55 GMT-500 2018
- Sun Jan 06 19:38:55 GMT-500 2018

Relational Database Service (RDS)

- RDBMS as a service
- Automated setup and patching
- Managed backups
- Point-in-time snapshots



Point-in-time snapshots



RDS > Instances > mytestdb > Restore to point in time

Launch DB Instance

You are creating a new DB instance from a source DB instance at a specified time. This new DB instance will have the default DB security group and DB parameter groups.

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Restore time

Point in time to restore from

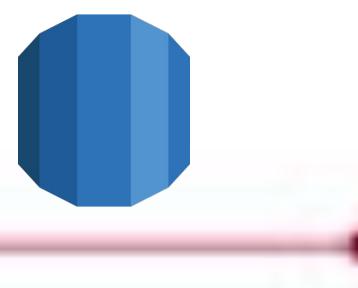
Latest restorable time
January 6, 2018 at 12:18:32 PM UTC-5

Custom
Specify a custom date and time to restore from

Custom Date Custom Time

MMMM d, y 00 ▾ : 00 ▾ : 00 ▾ UTC-5

Point-in-time snapshots



RDS > Instances > mytestdb > Restore to point in time

Launch DB Instance

You are creating a new DB instance from a source DB instance at a specified time. This new DB instance will have the default DB security group and DB parameter groups.

This feature is currently supported for InnoDB storage engine only. If you are using MyISAM, refer to details [here](#).

Point in time to restore from

Latest restorable time
January 6, 2018 at 12:18:32 PM UTC-5

Custom
Specify a custom date and time to restore from

MM/DD/YYYY HH : MM : SS AM/PM UTC-5

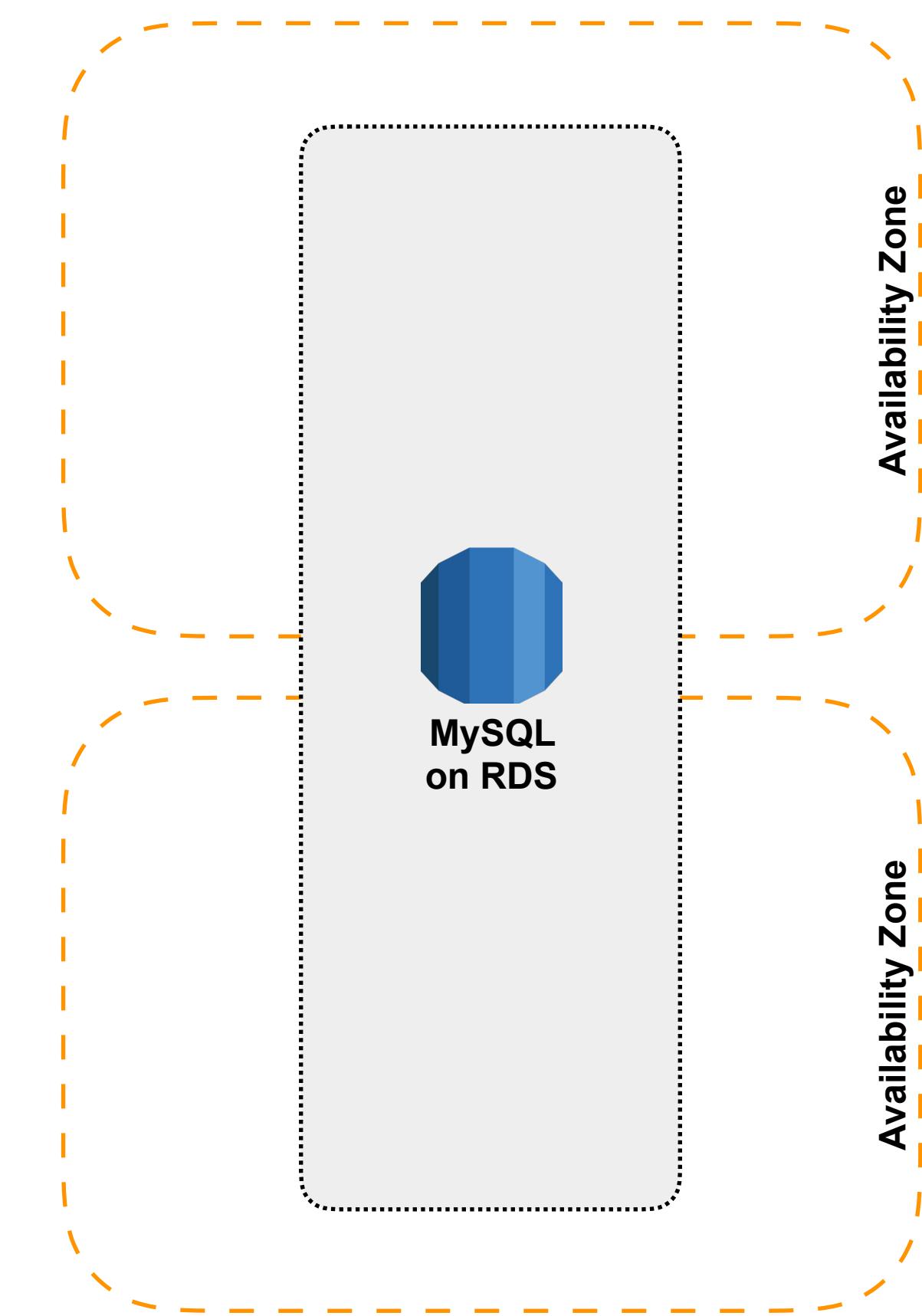
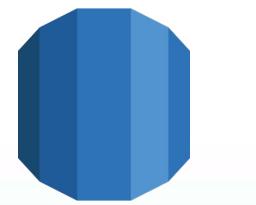
A screenshot of the AWS RDS 'Launch DB Instance' wizard. The 'Point in time to restore from' section is highlighted. It shows two options: 'Latest restorable time' (which is selected) and 'Custom'. Under 'Custom', there is a note: 'Specify a custom date and time to restore from'. Below this, there are dropdown menus for selecting the date (MM/DD/YYYY), time (HH : MM : SS), and timezone (UTC-5). The 'Custom' option is selected, indicated by a blue dot next to its radio button.

Relational Database Service (RDS)

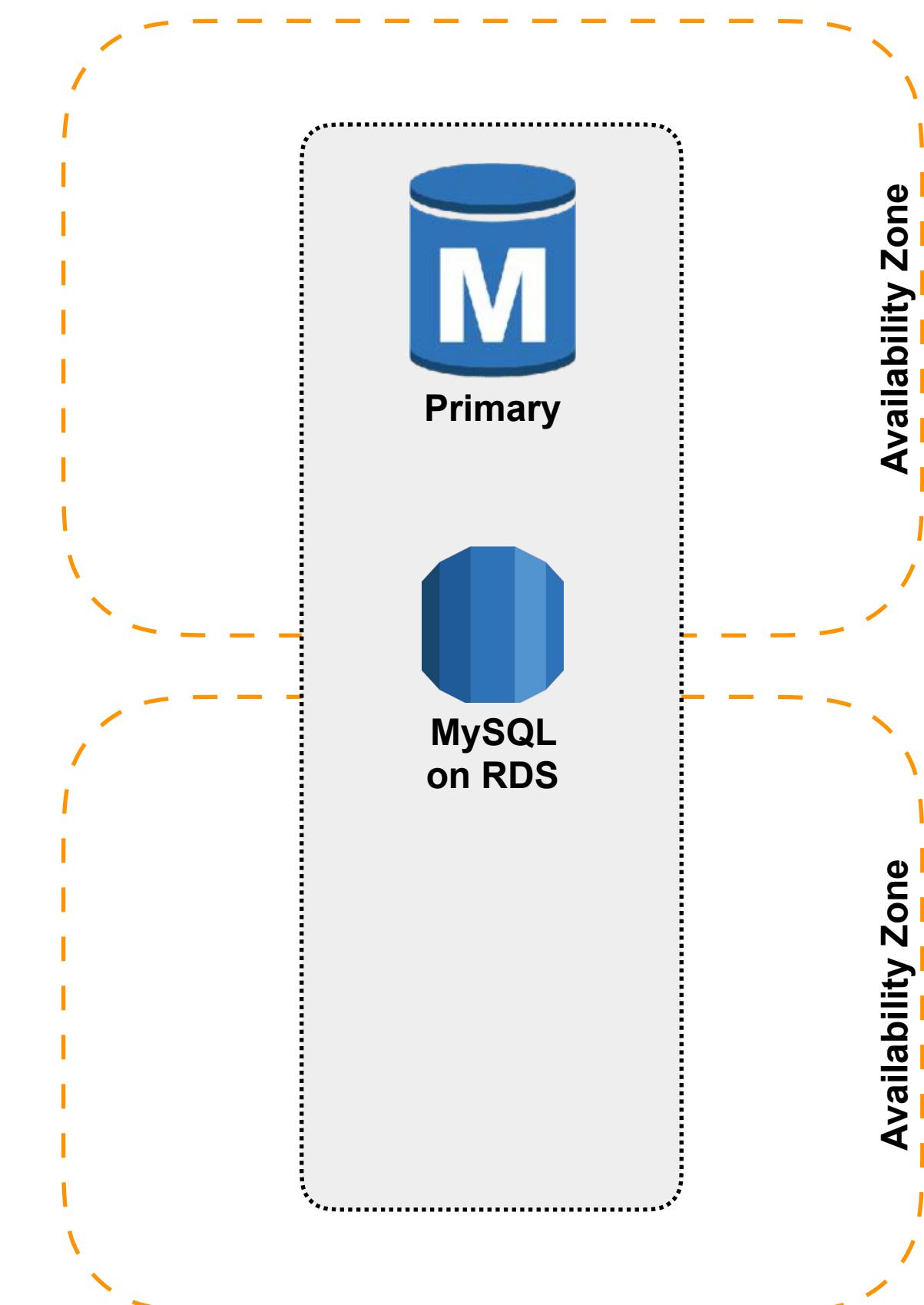
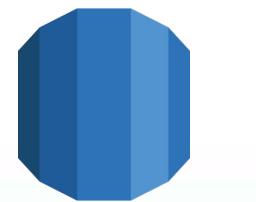
- RDBMS as a service
- Automated setup and patching
- Managed backups
- Point-in-time snapshots
- Zero downtime scaling and fast recovery from outages



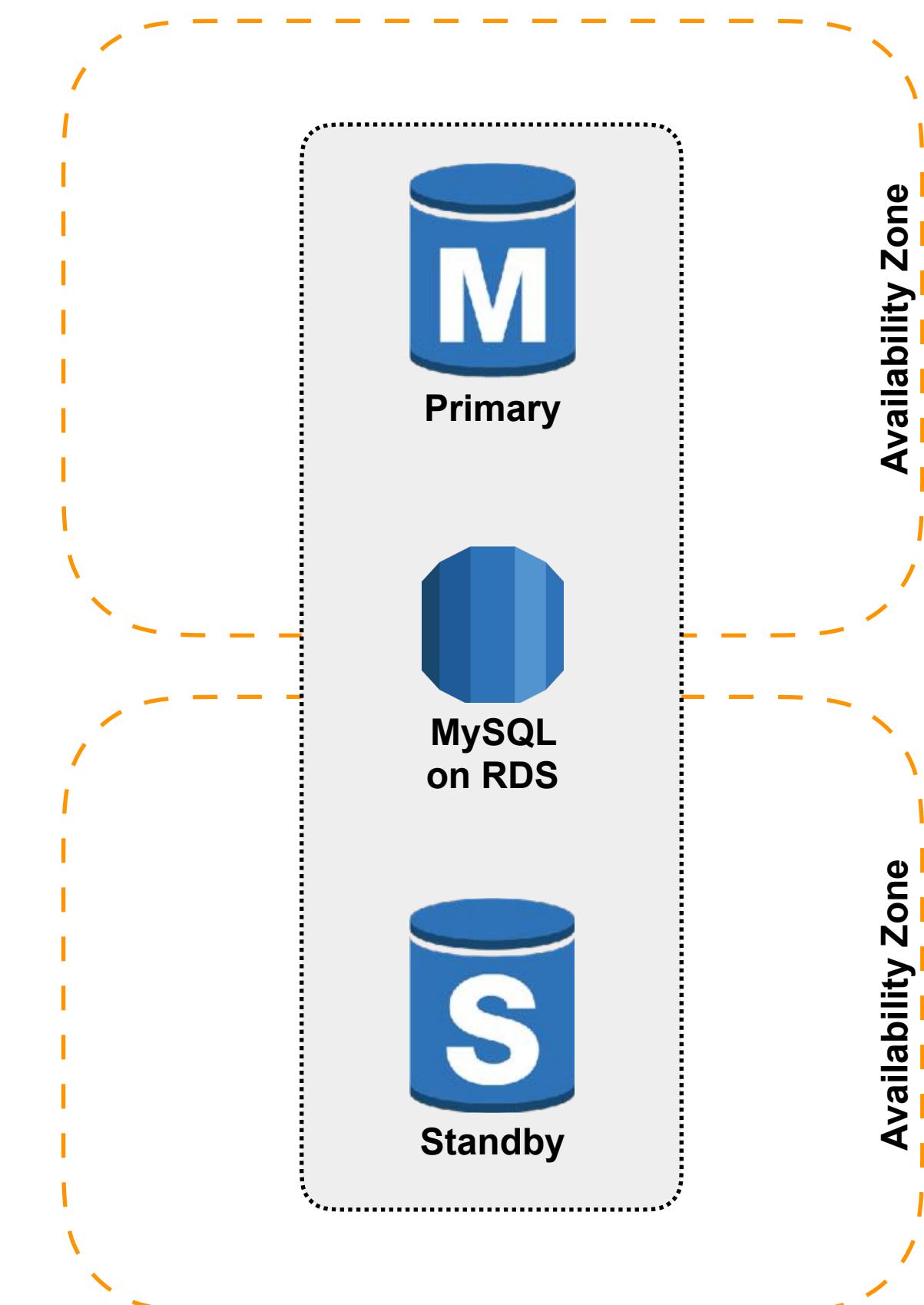
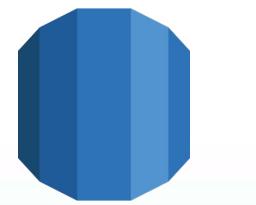
Zero downtime scaling and fast recovery



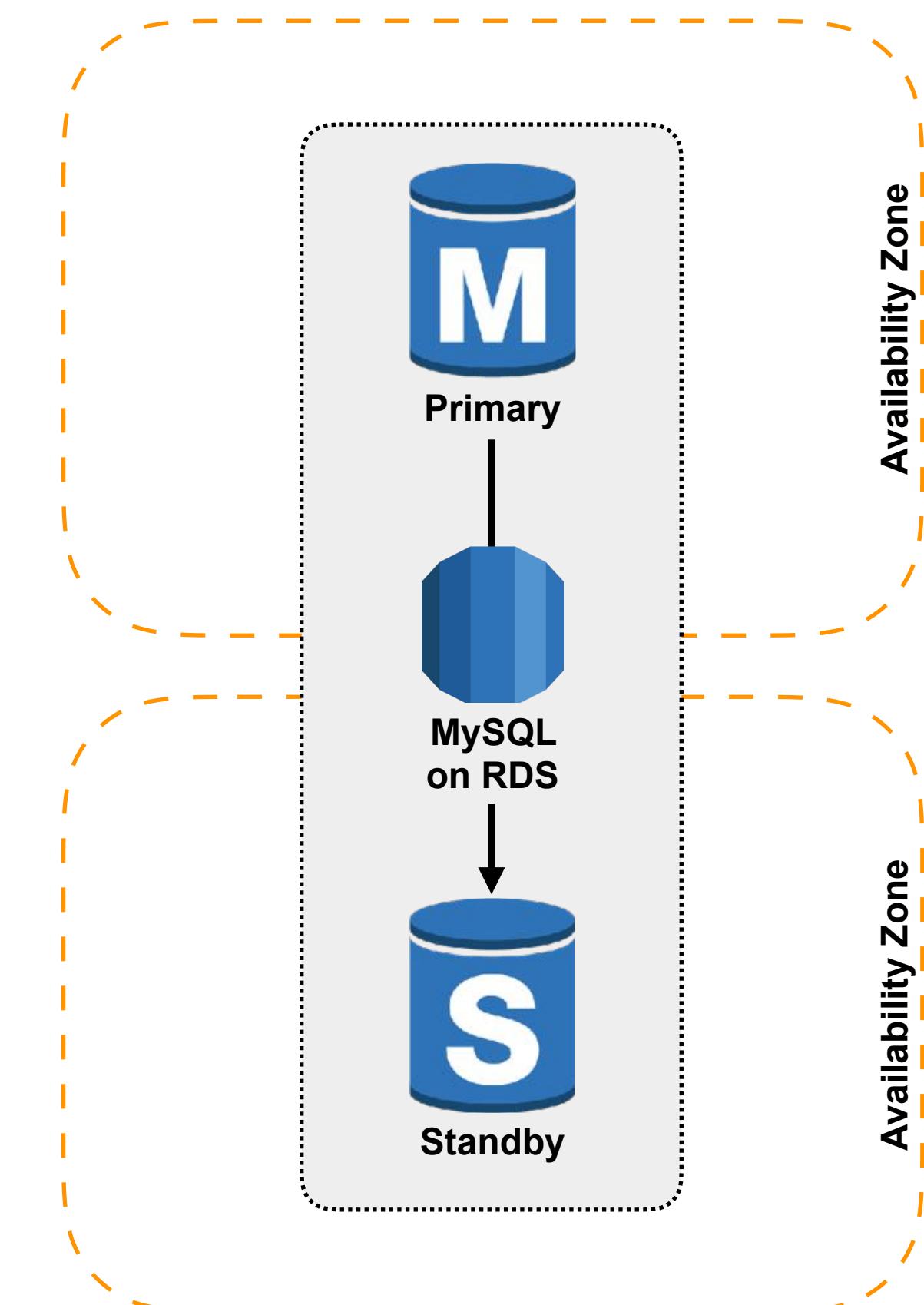
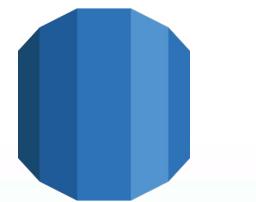
Zero downtime scaling and fast recovery



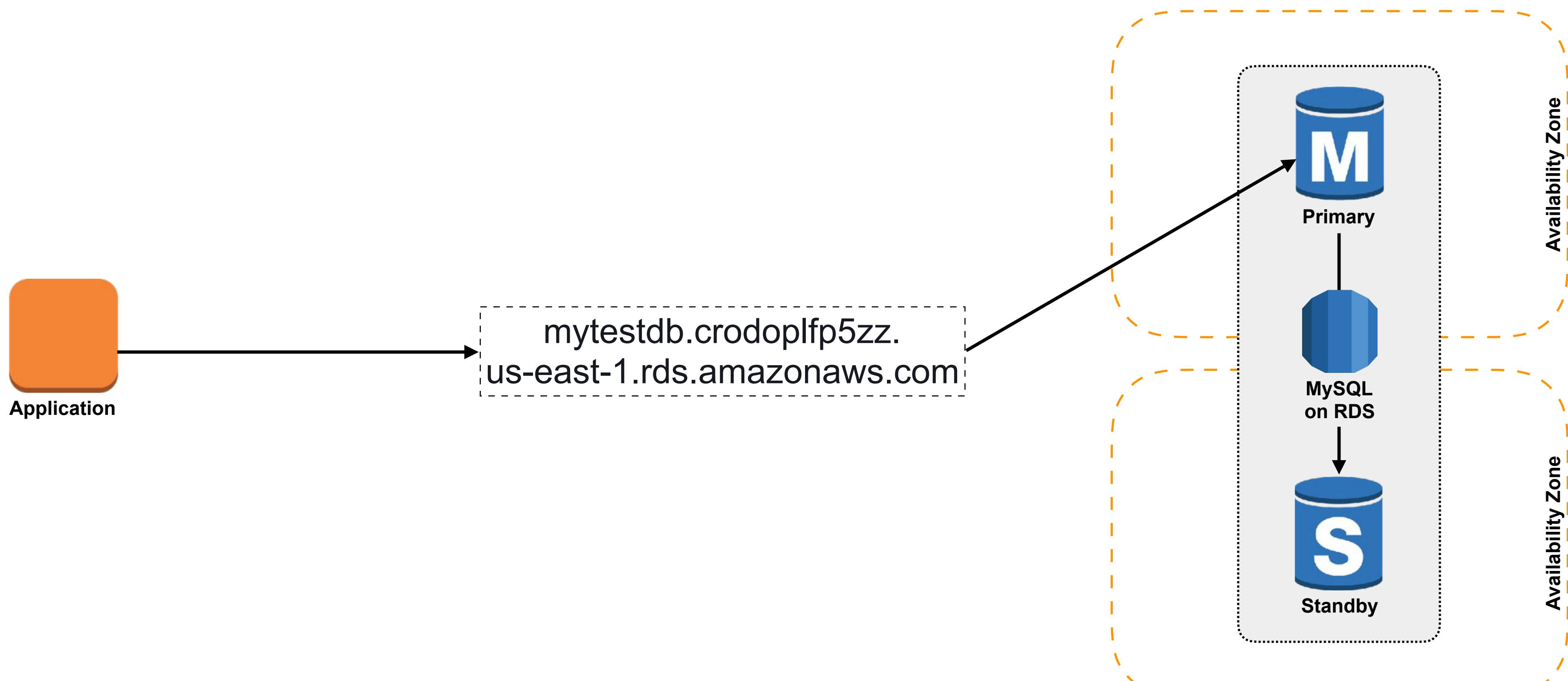
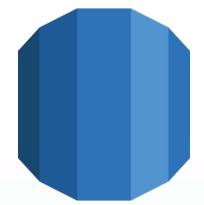
Zero downtime scaling and fast recovery



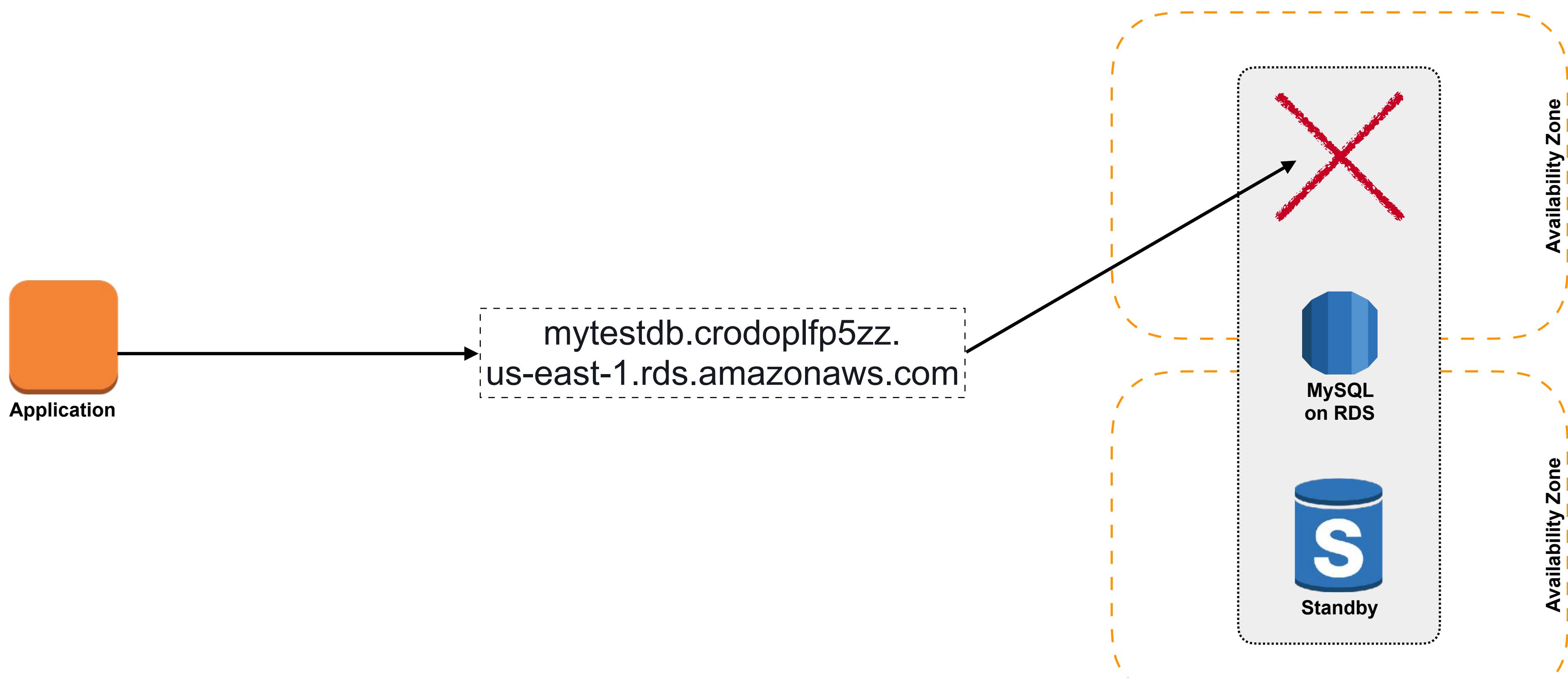
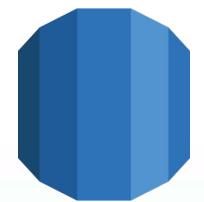
Zero downtime scaling and fast recovery



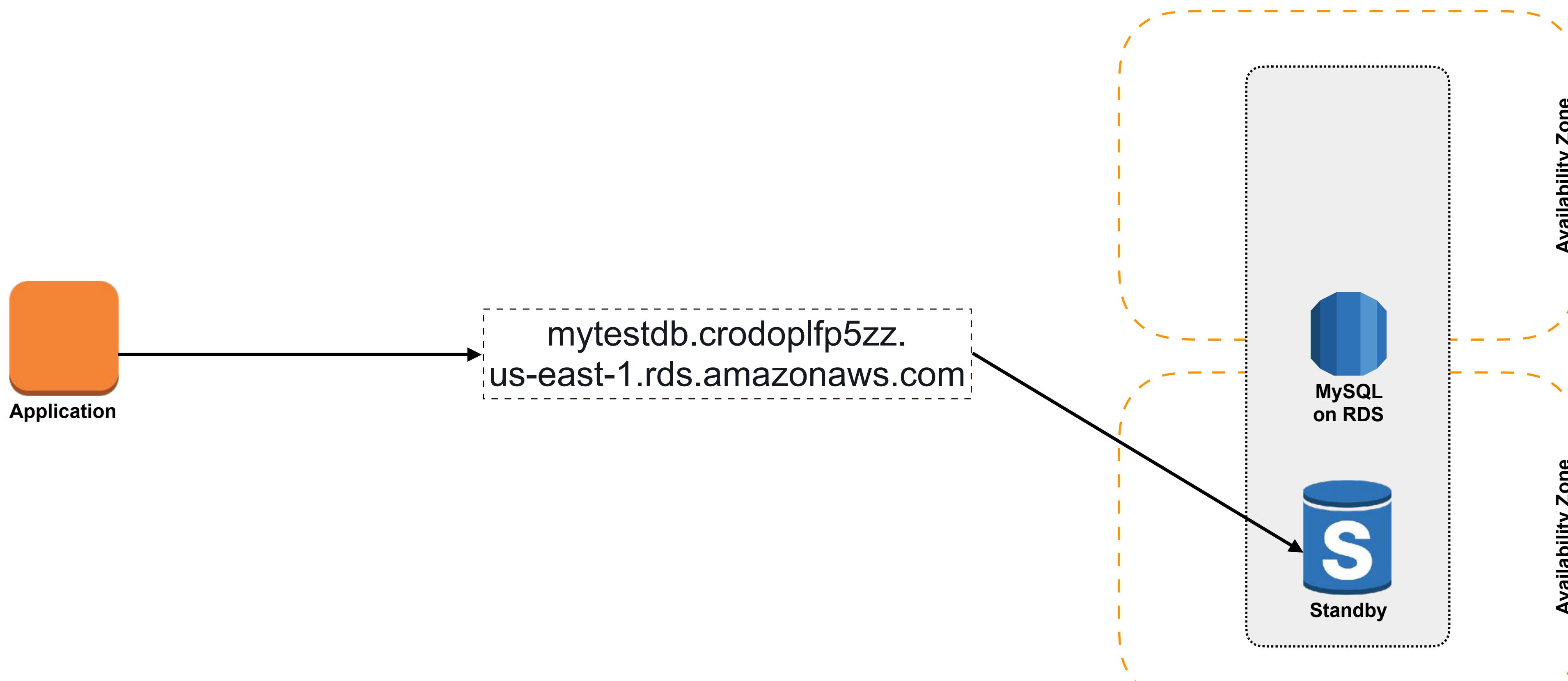
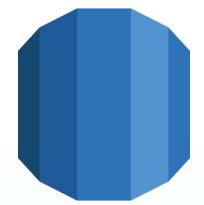
Zero downtime scaling and fast recovery



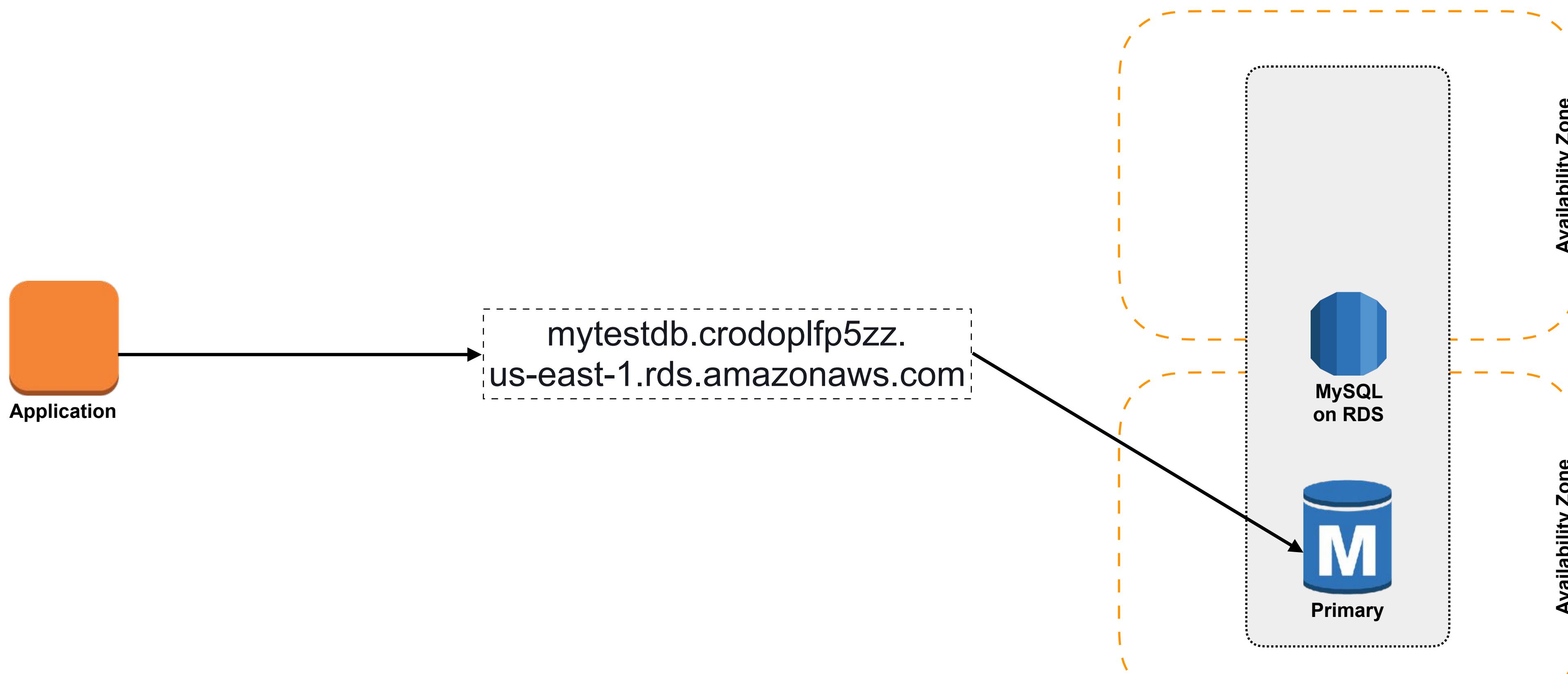
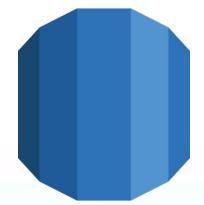
Zero downtime scaling and fast recovery



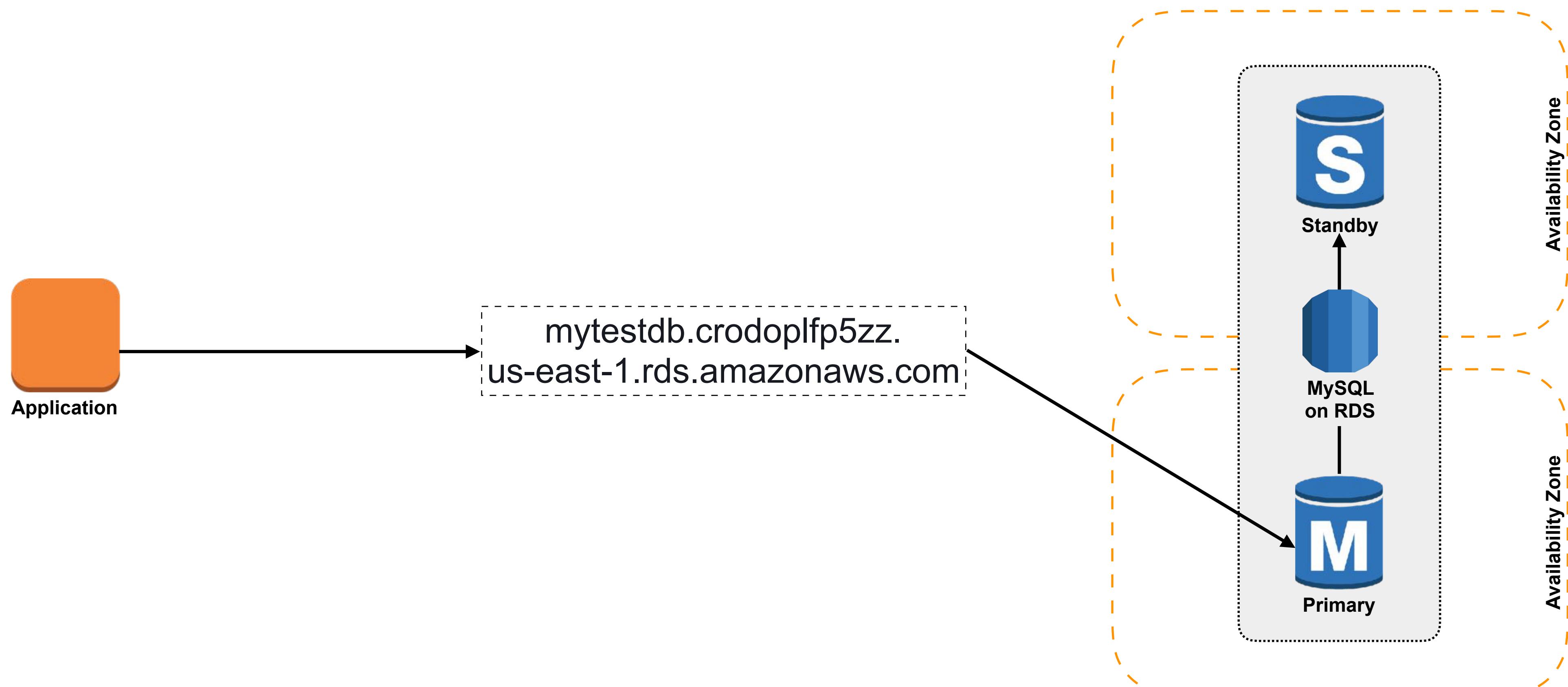
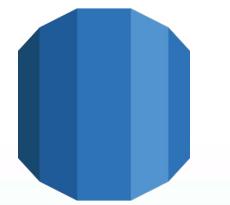
Zero downtime scaling and fast recovery



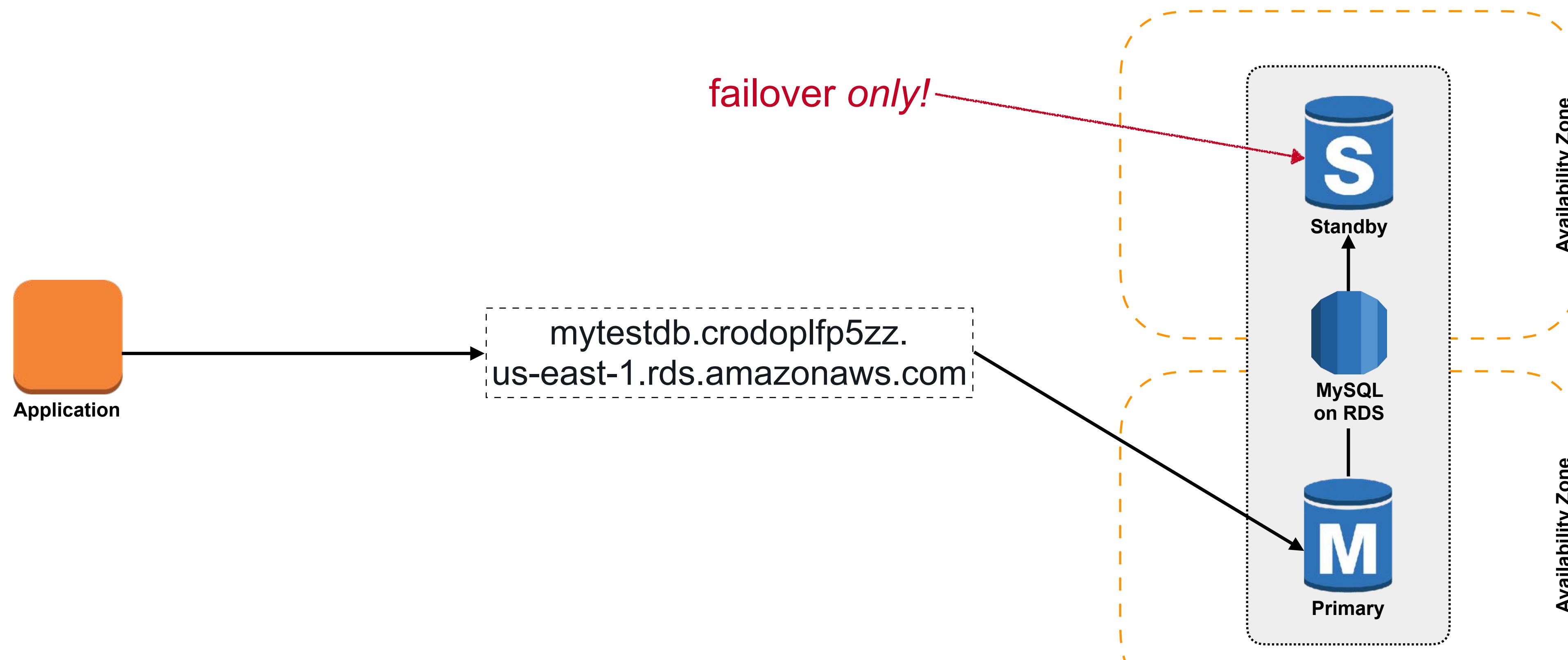
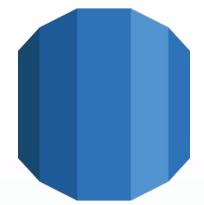
Zero downtime scaling and fast recovery



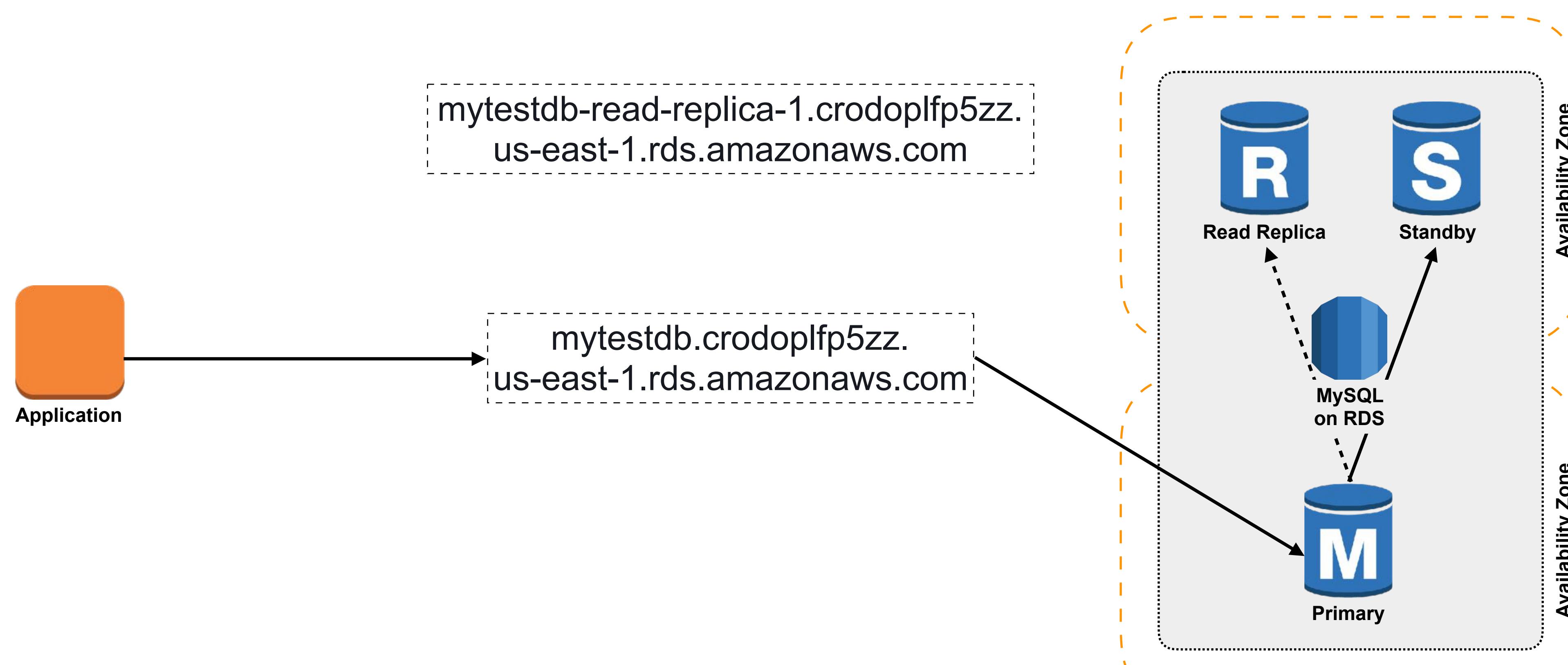
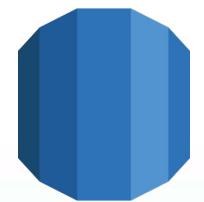
Zero downtime scaling and fast recovery



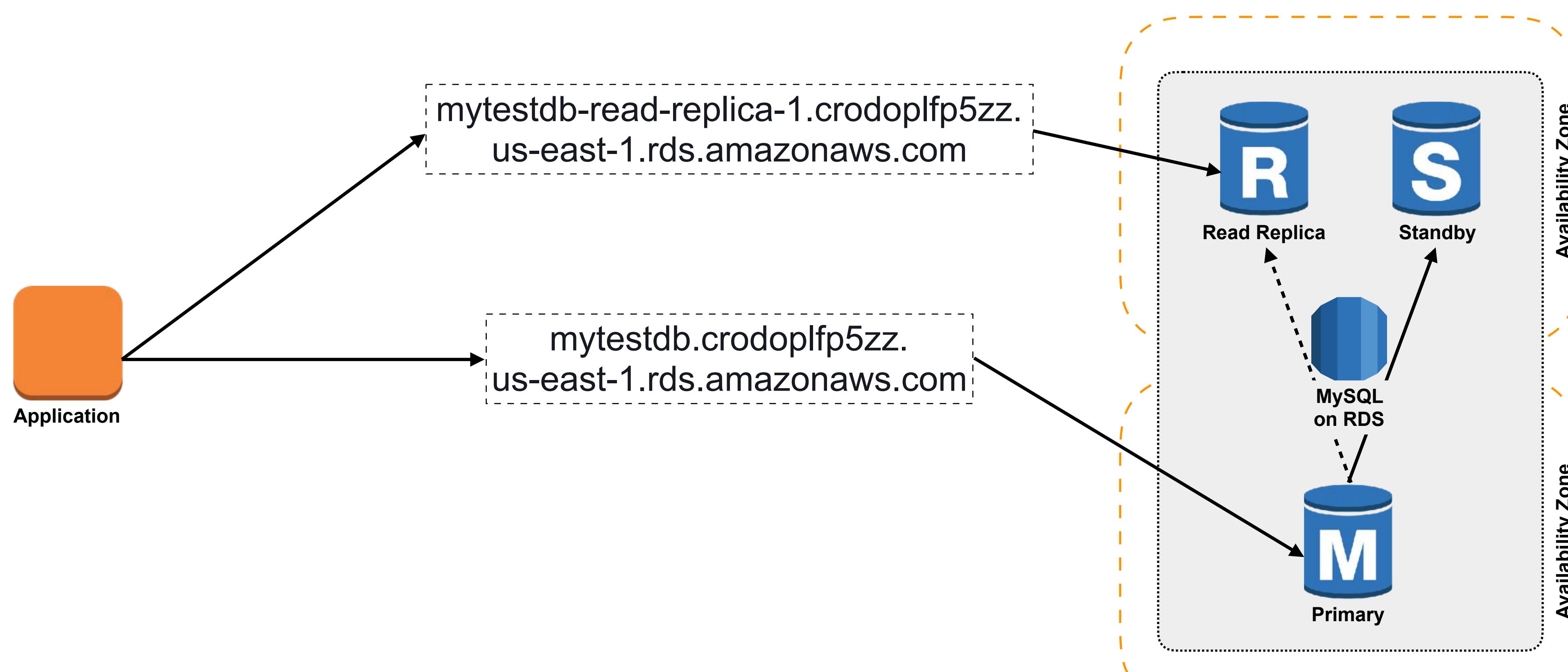
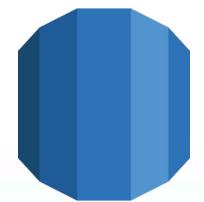
Zero downtime scaling and fast recovery



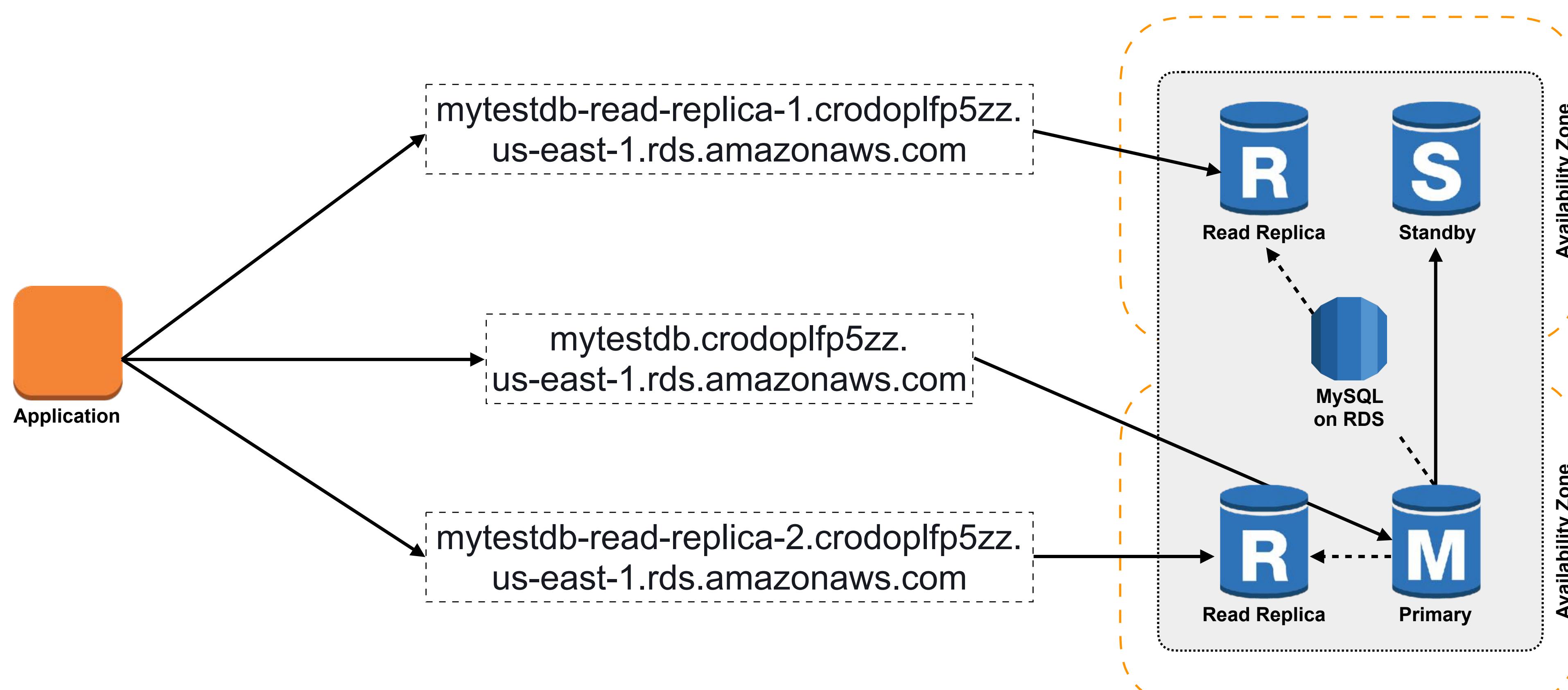
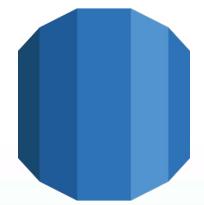
Zero downtime scaling and fast recovery



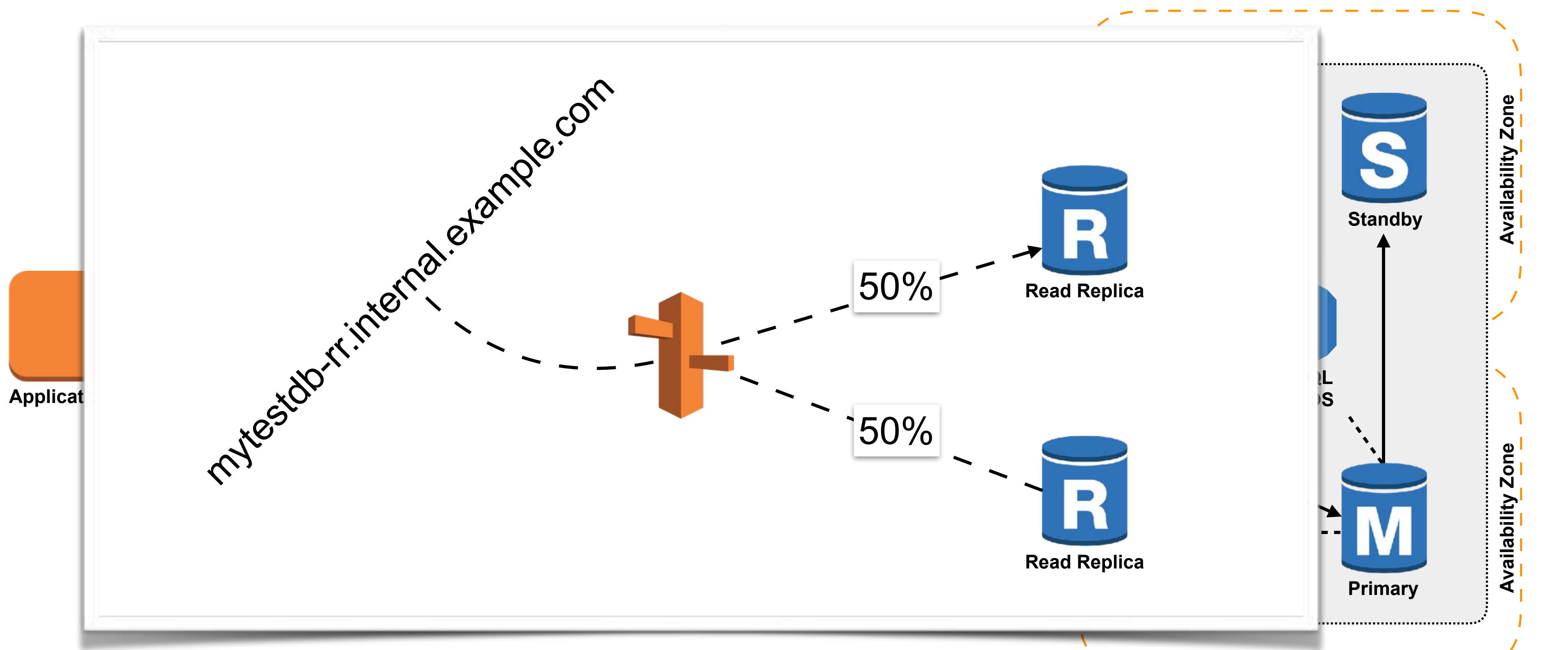
Zero downtime scaling and fast recovery



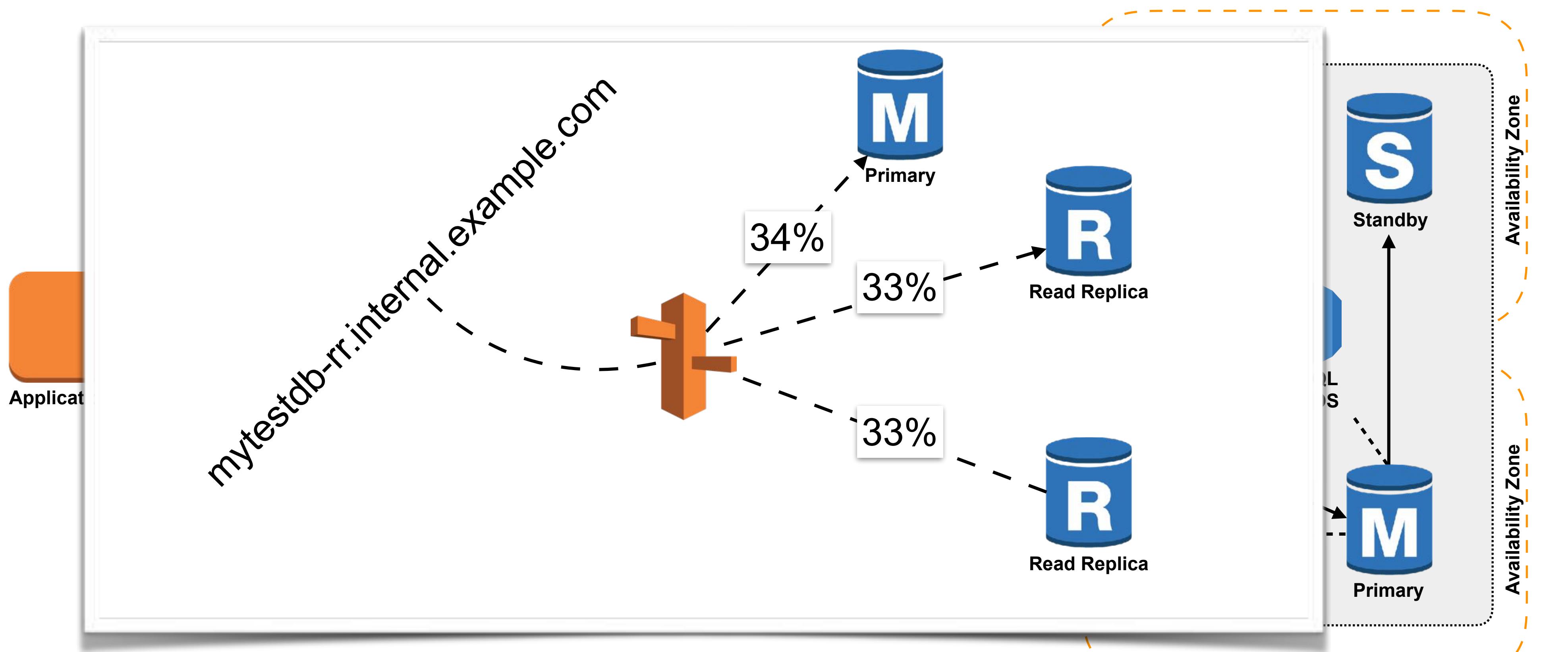
Zero downtime scaling and fast recovery



Zero downtime scaling and fast recovery



Zero downtime scaling and fast recovery



Relational Database Service (RDS)

- RDBMS as a service
- Automated setup and patching
- Managed backups
- Point-in-time snapshots
- Zero downtime scaling and fast recovery from outages



Relational Database Service (RDS)

Amazon Aurora

- MySQL and PostgreSQL compatible



Relational Database Service (RDS)

Amazon Aurora

- MySQL and PostgreSQL compatible
 - Pay-as-you-go on-the-fly storage



Relational Database Service (RDS)

Amazon Aurora

- MySQL and PostgreSQL compatible
 - Pay-as-you-go on-the-fly storage
 - Up to 15 read replicas for failover



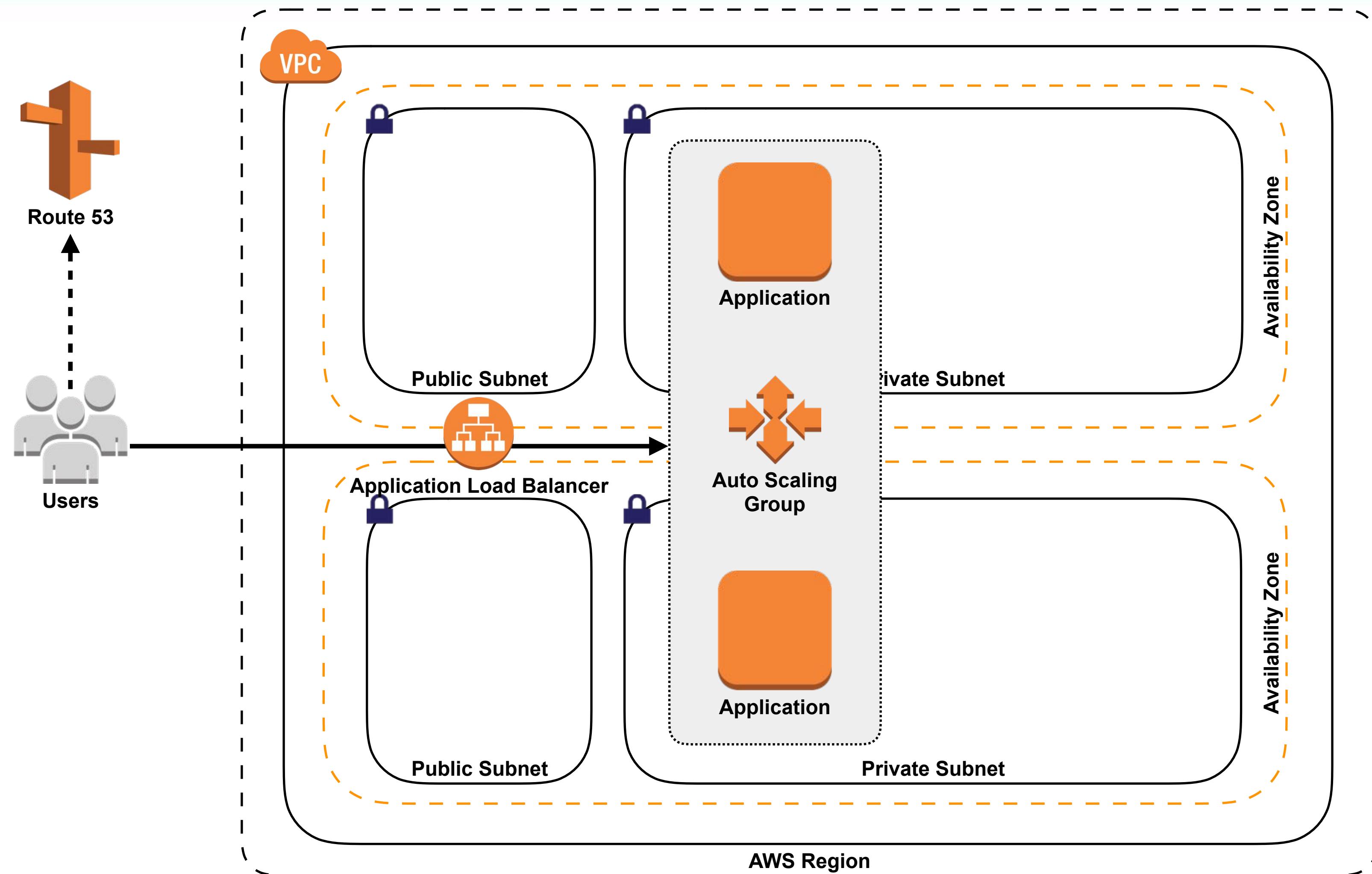
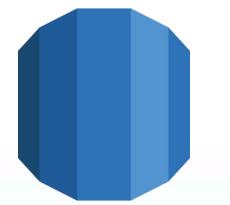
Relational Database Service (RDS)

Amazon Aurora

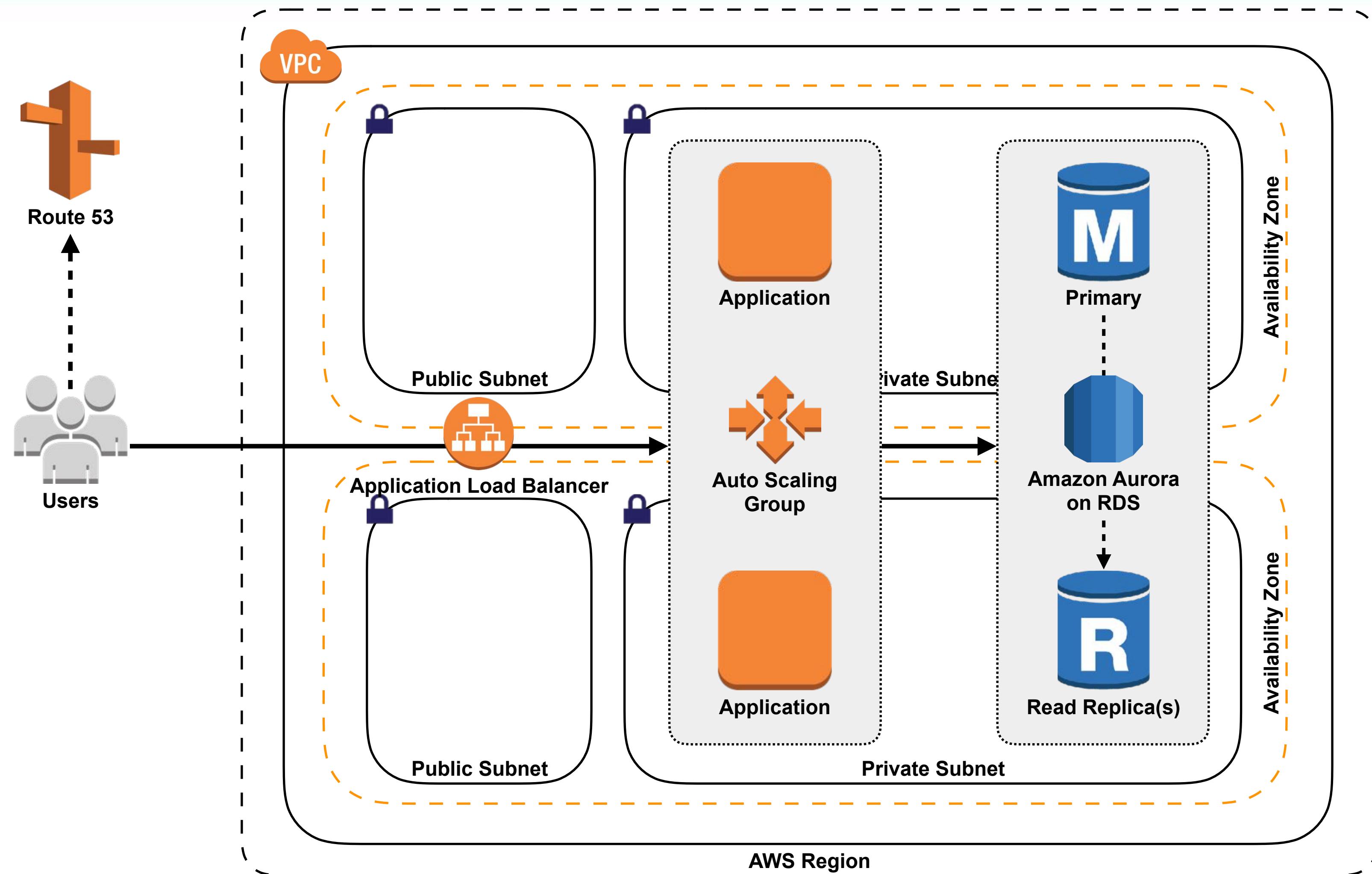
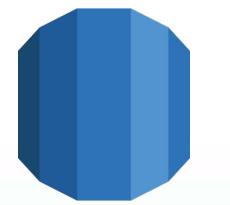
- MySQL and PostgreSQL compatible
- Pay-as-you-go on-the-fly storage
- Up to 15 read replicas for failover
- Very low replication lag



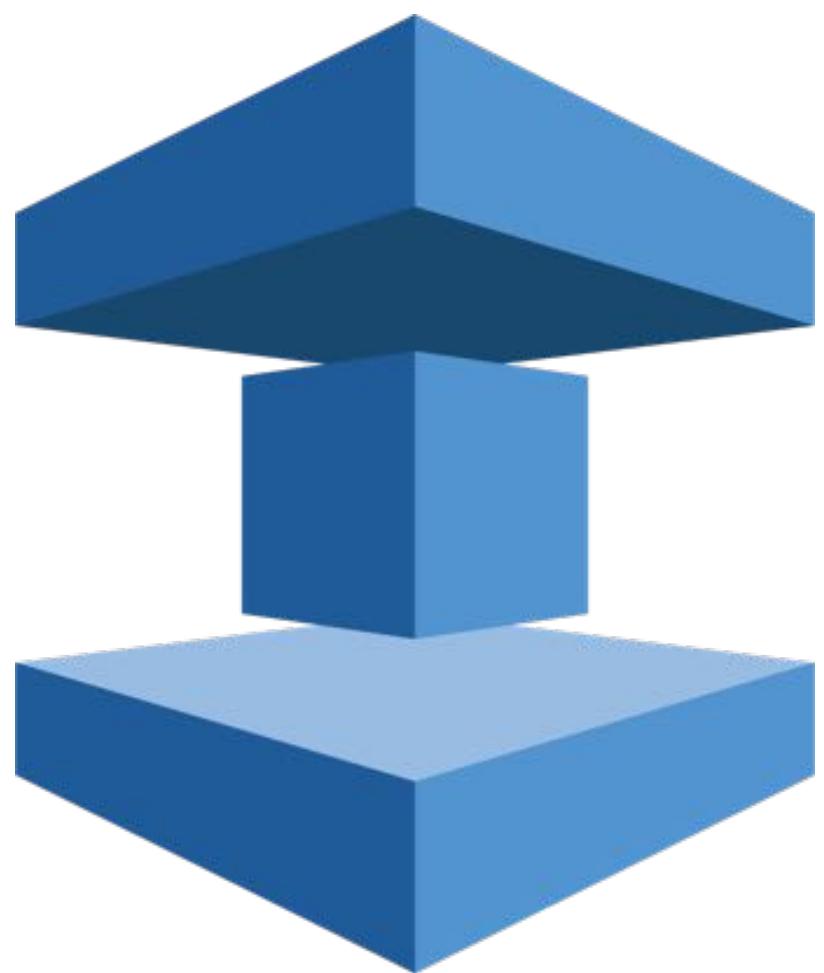
Relational Database Service (RDS)



Relational Database Service (RDS)



Amazon ElastiCache



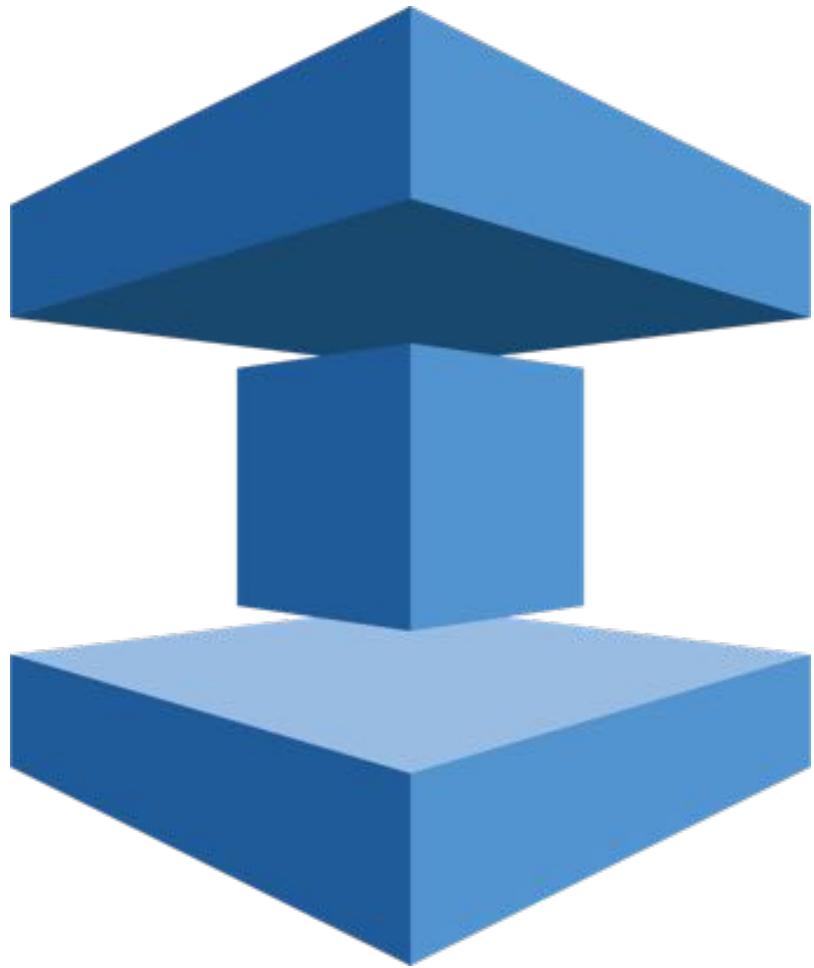
Amazon ElastiCache

- In-memory cache as a service



Amazon ElastiCache

- In-memory cache as a service
- Automated setup and patching



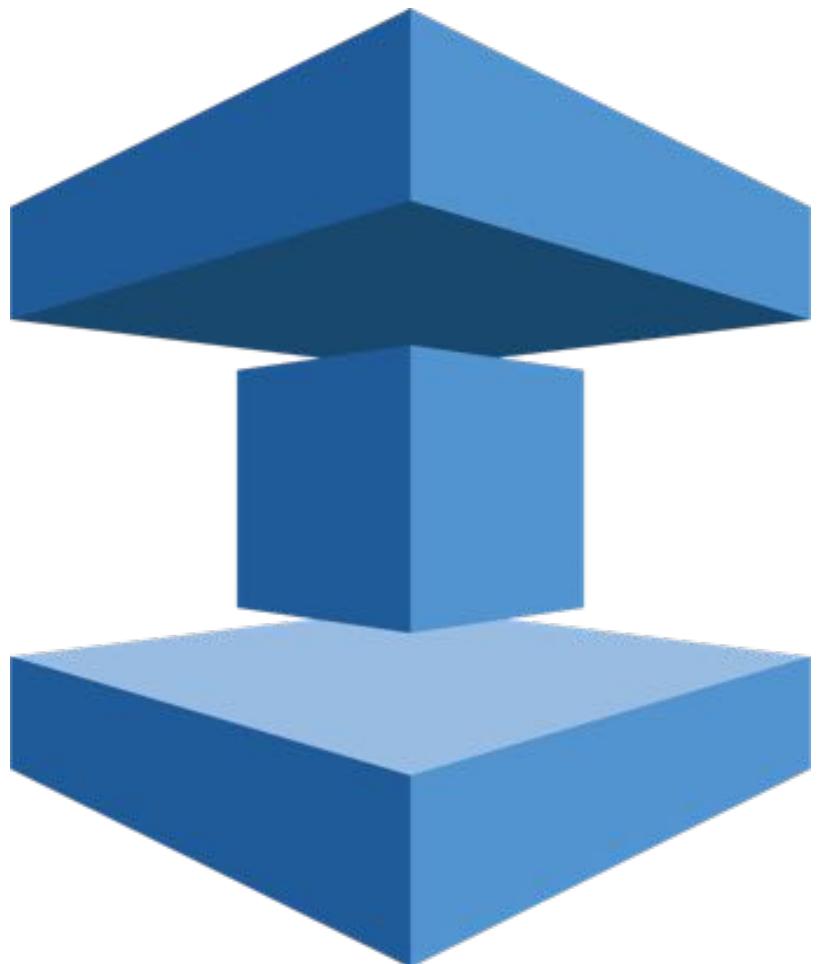
Amazon ElastiCache

- In-memory cache as a service
- Automated setup and patching
- Memcached or Redis



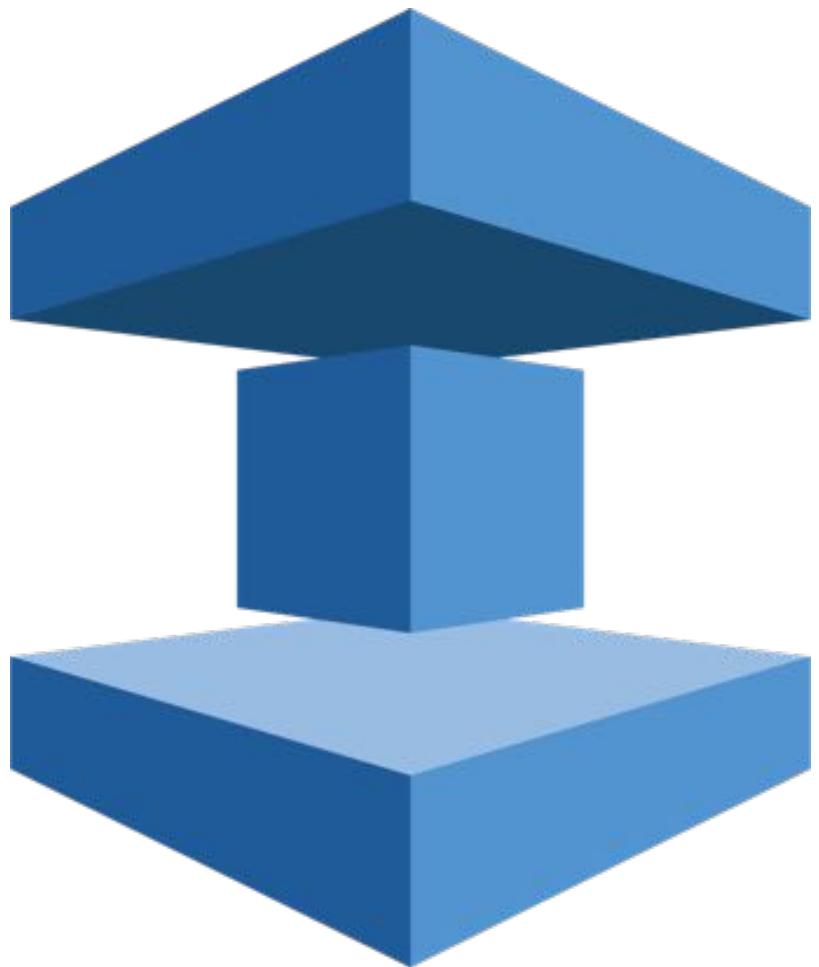
Amazon ElastiCache

- In-memory cache as a service
- Automated setup and patching
- Memcached or Redis
- **Fast!**

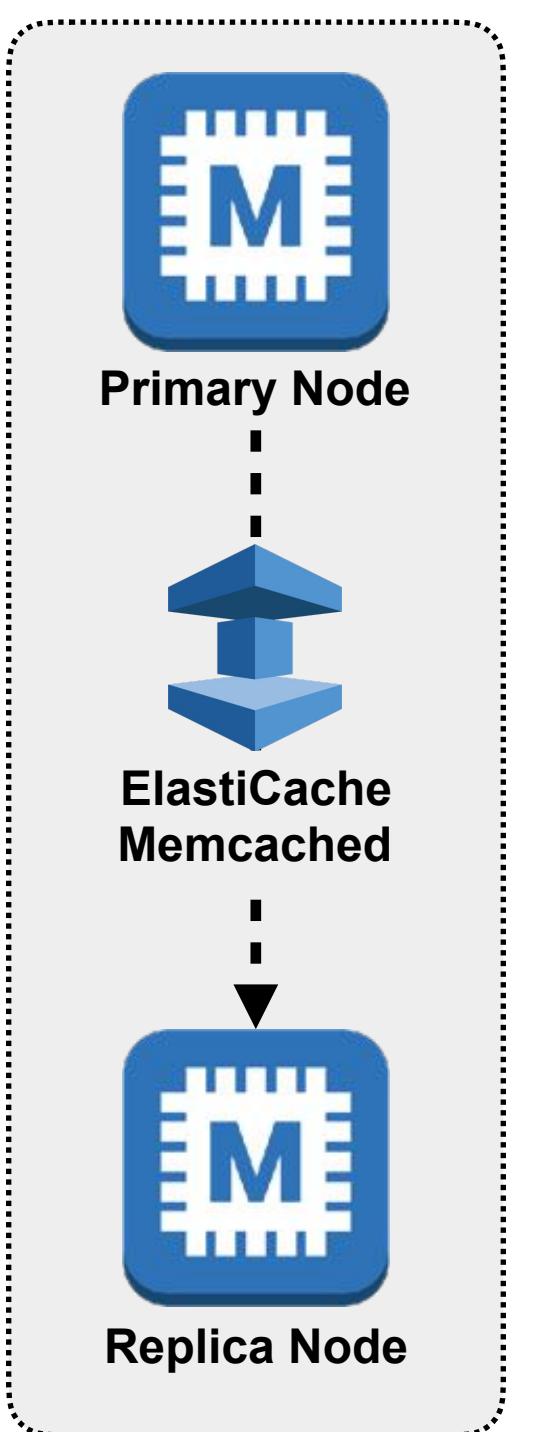


Amazon ElastiCache

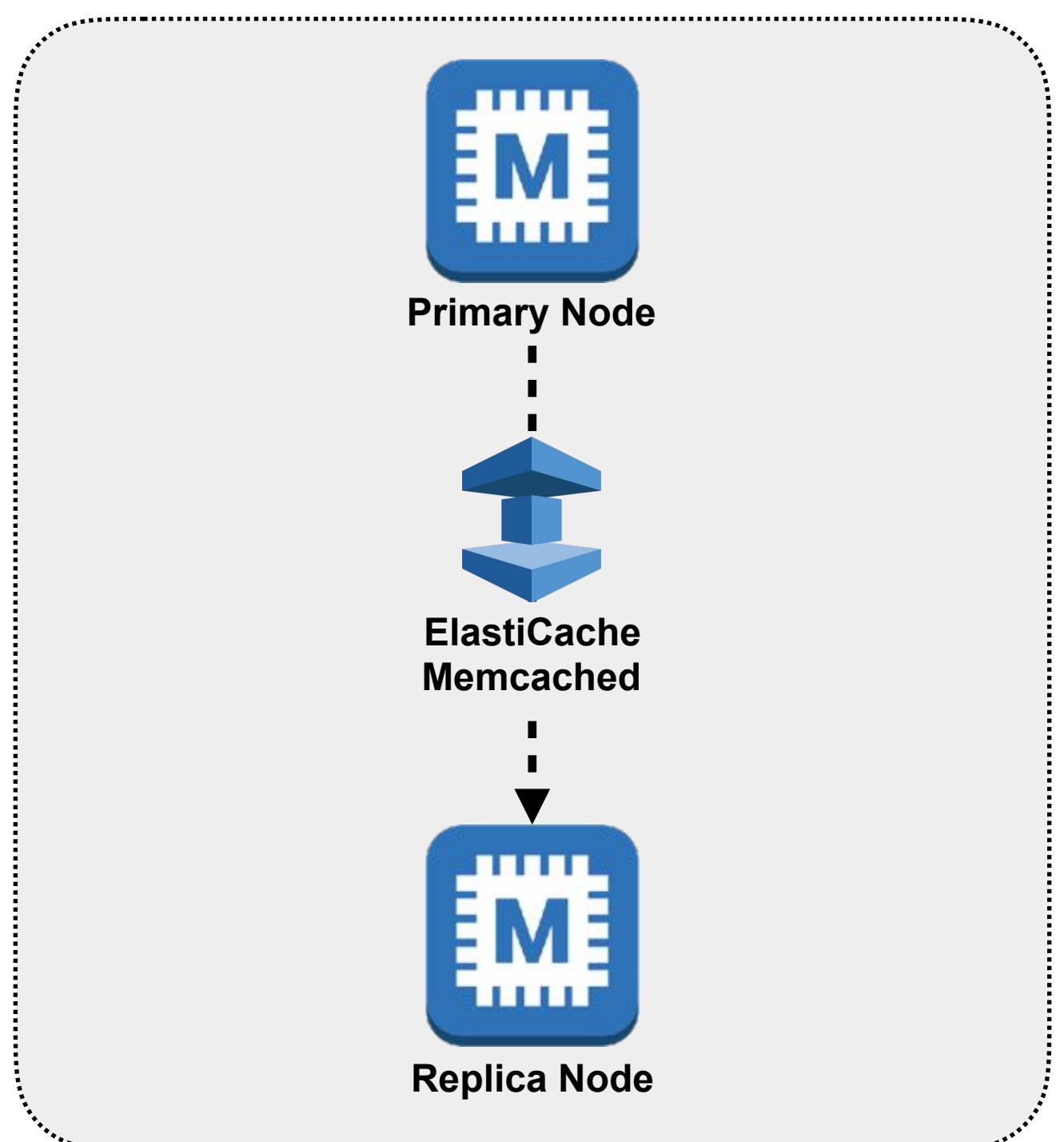
- In-memory cache as a service
- Automated setup and patching
- Memcached or Redis
- Fast!
- Easily scalable



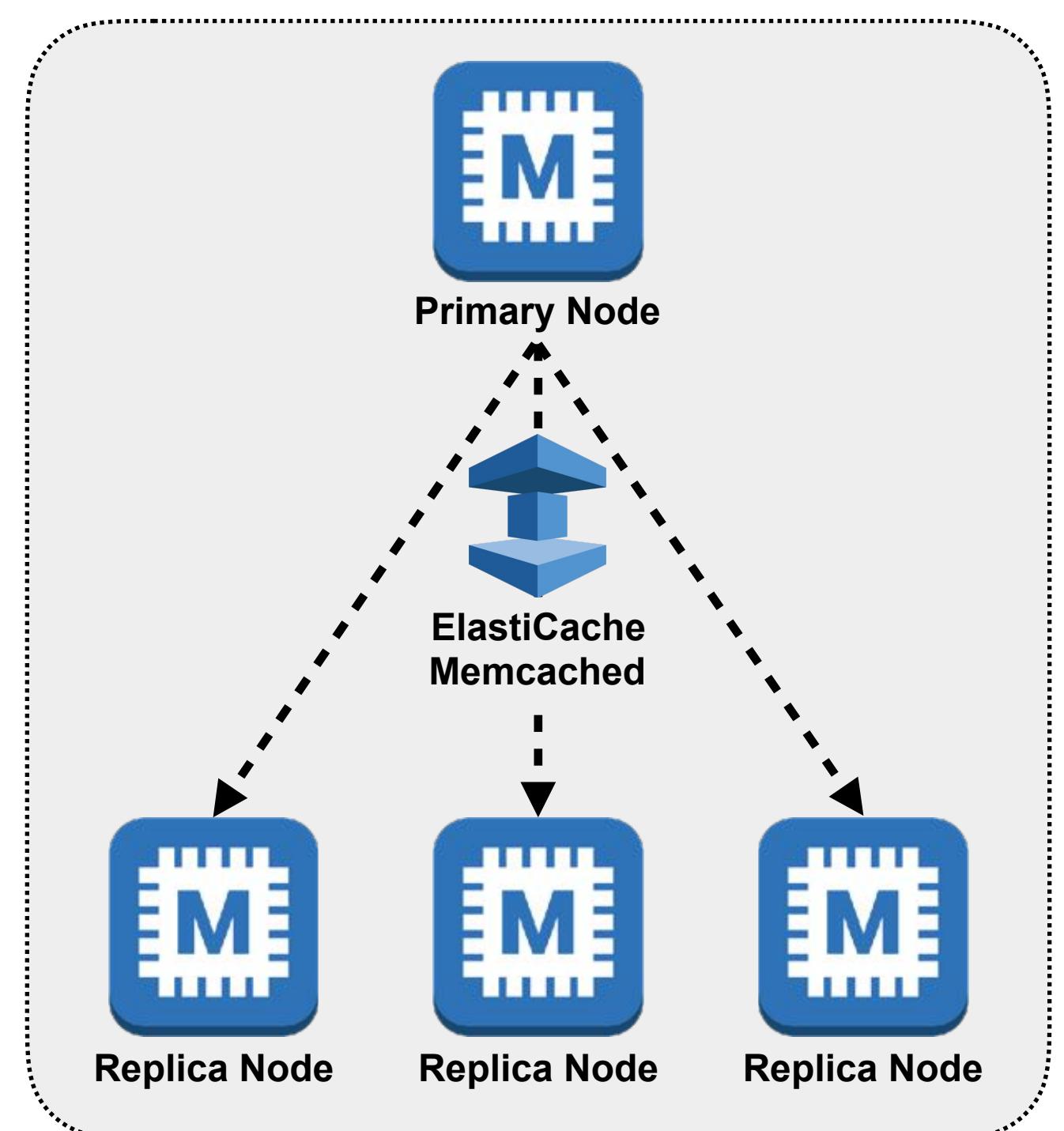
Easily scalable



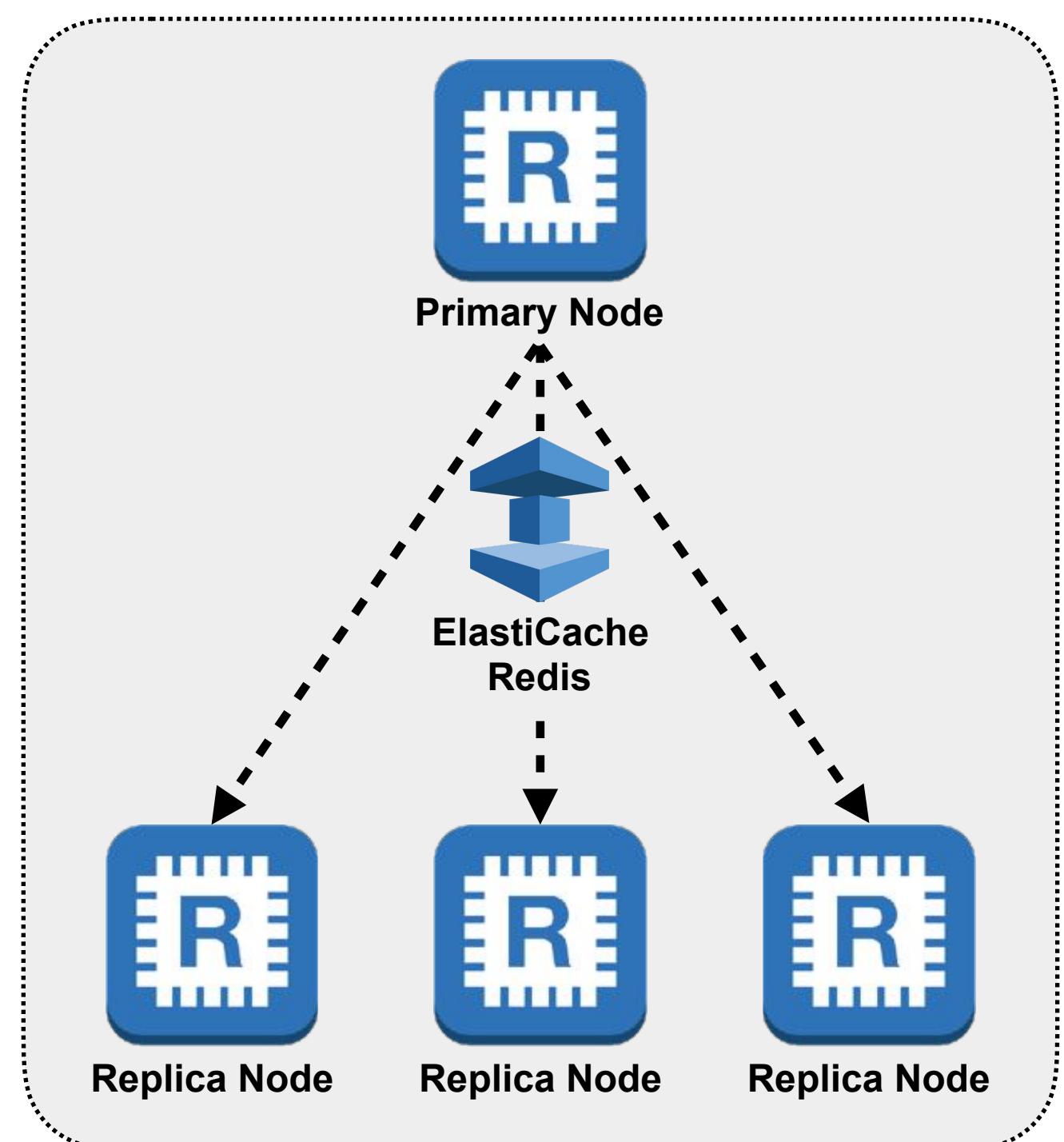
Easily scalable



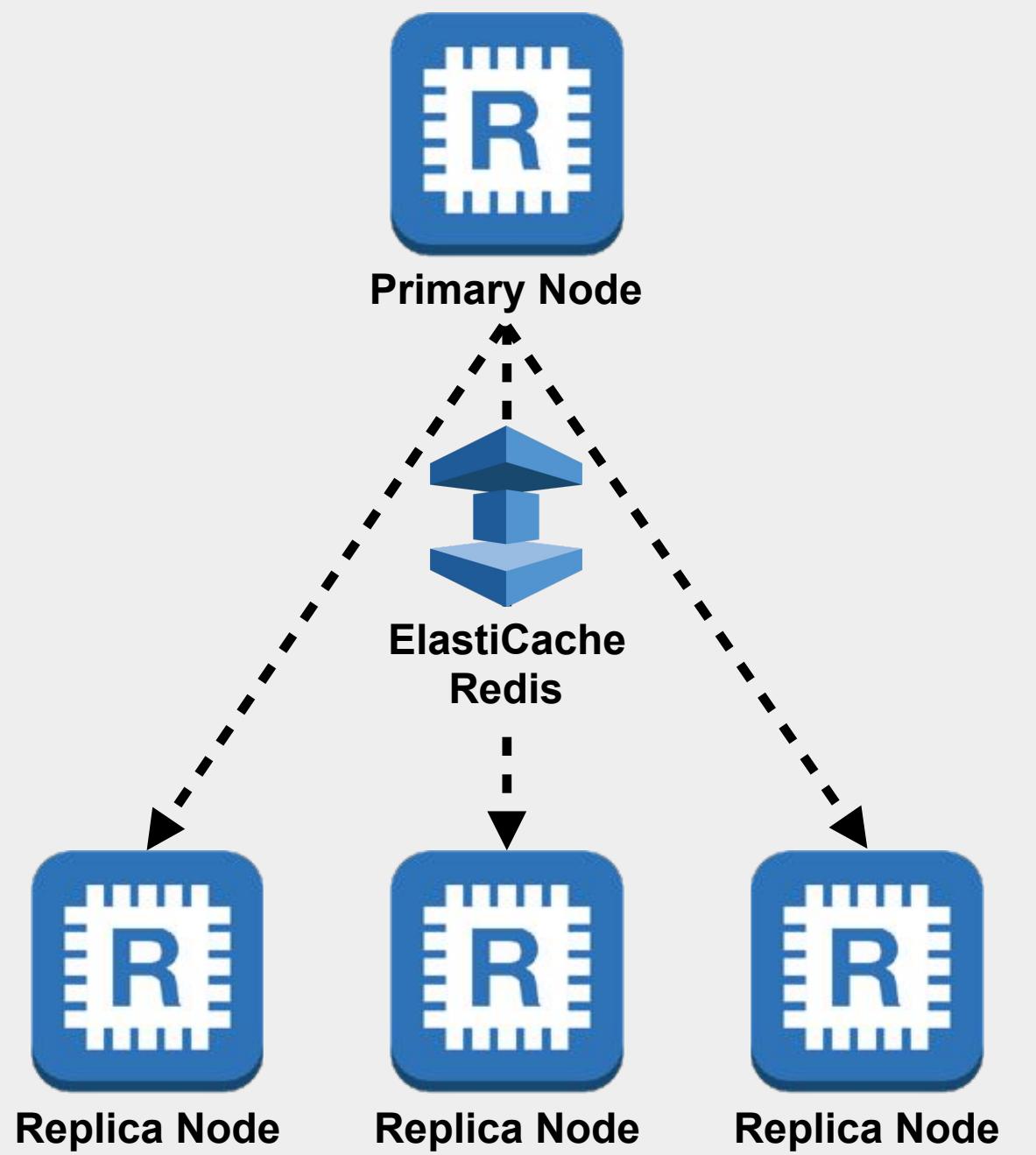
Easily scalable



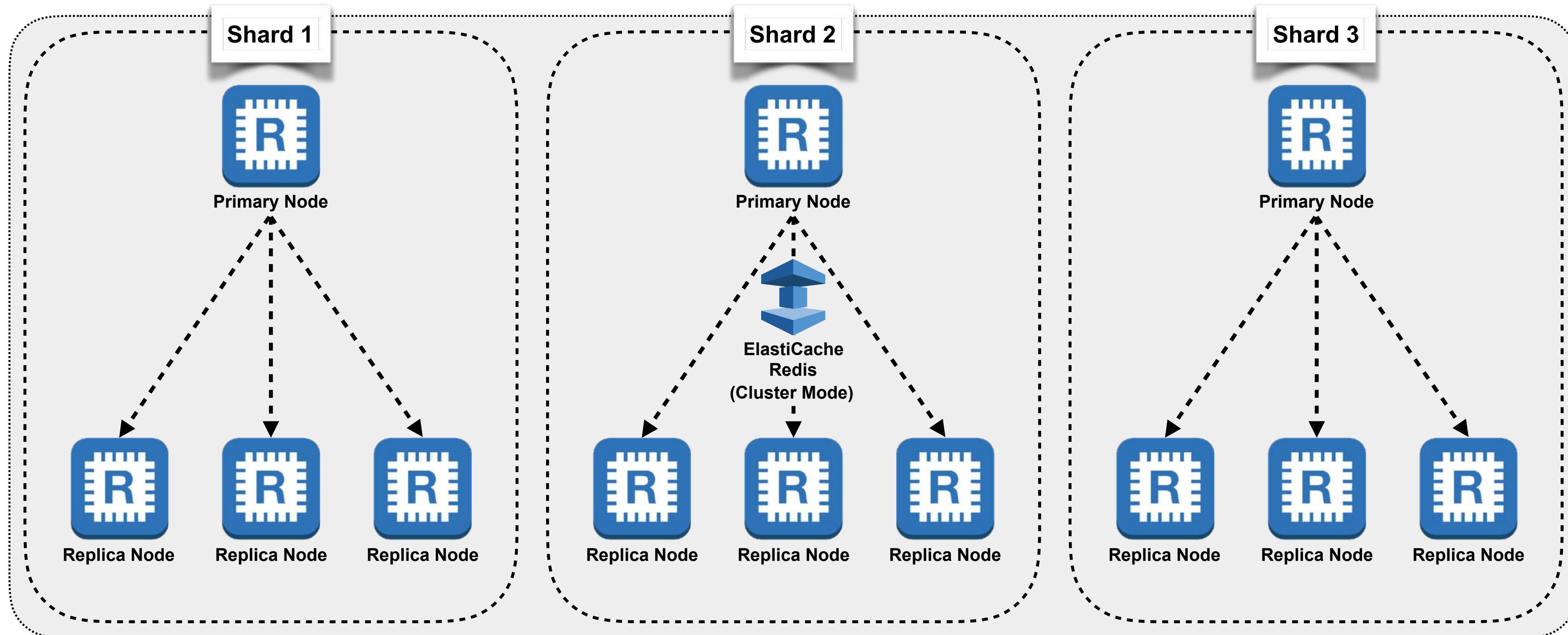
Easily scalable



Easily scalable

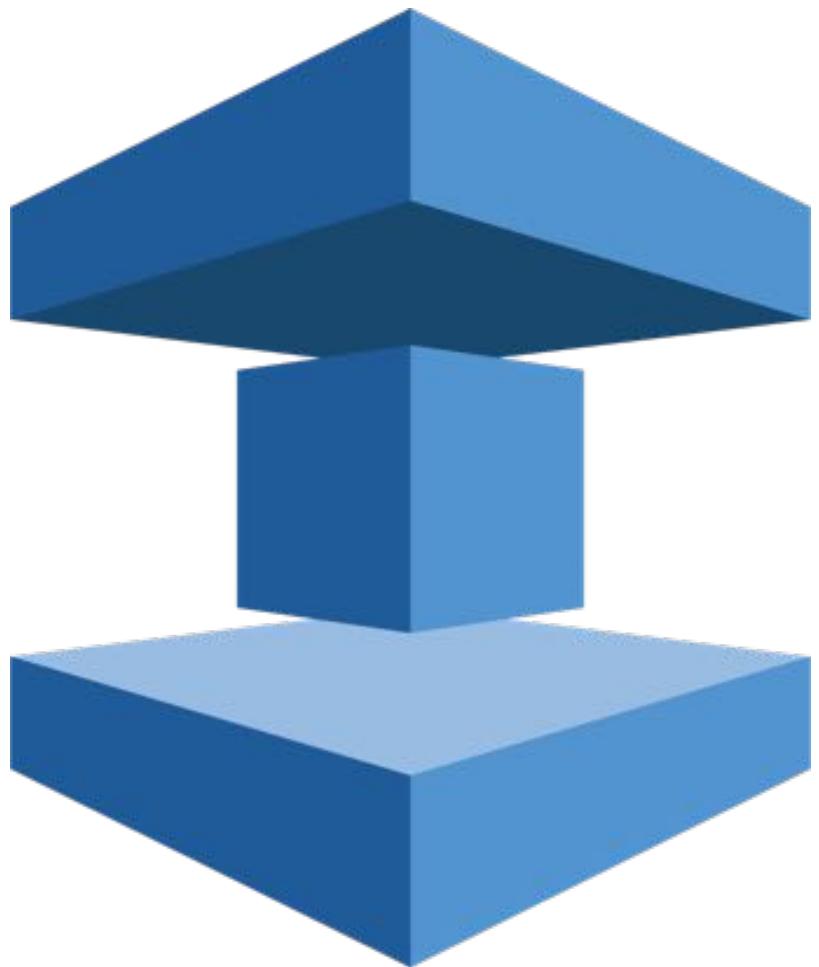


Easily scalable

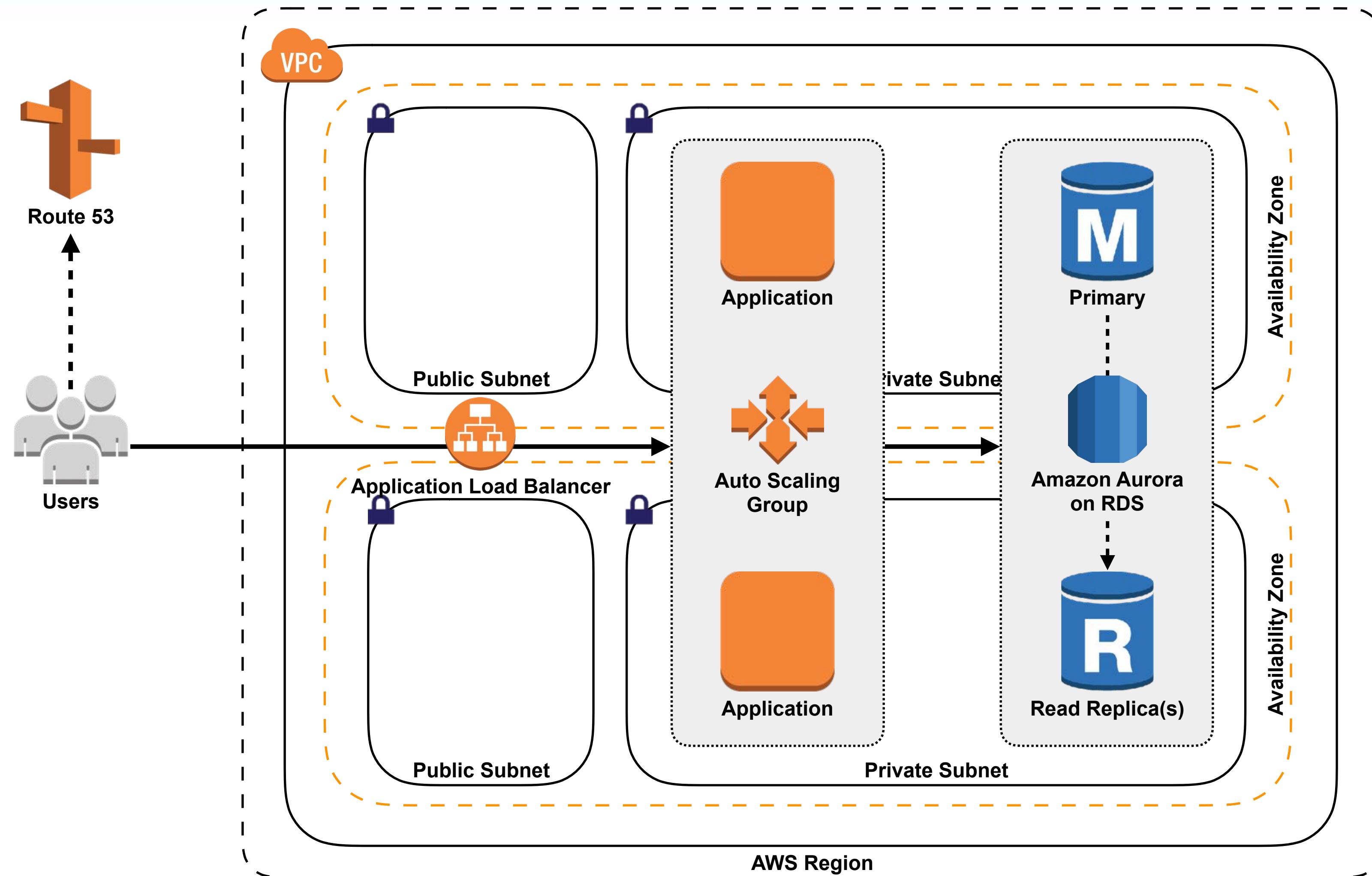
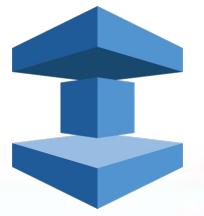


Amazon ElastiCache

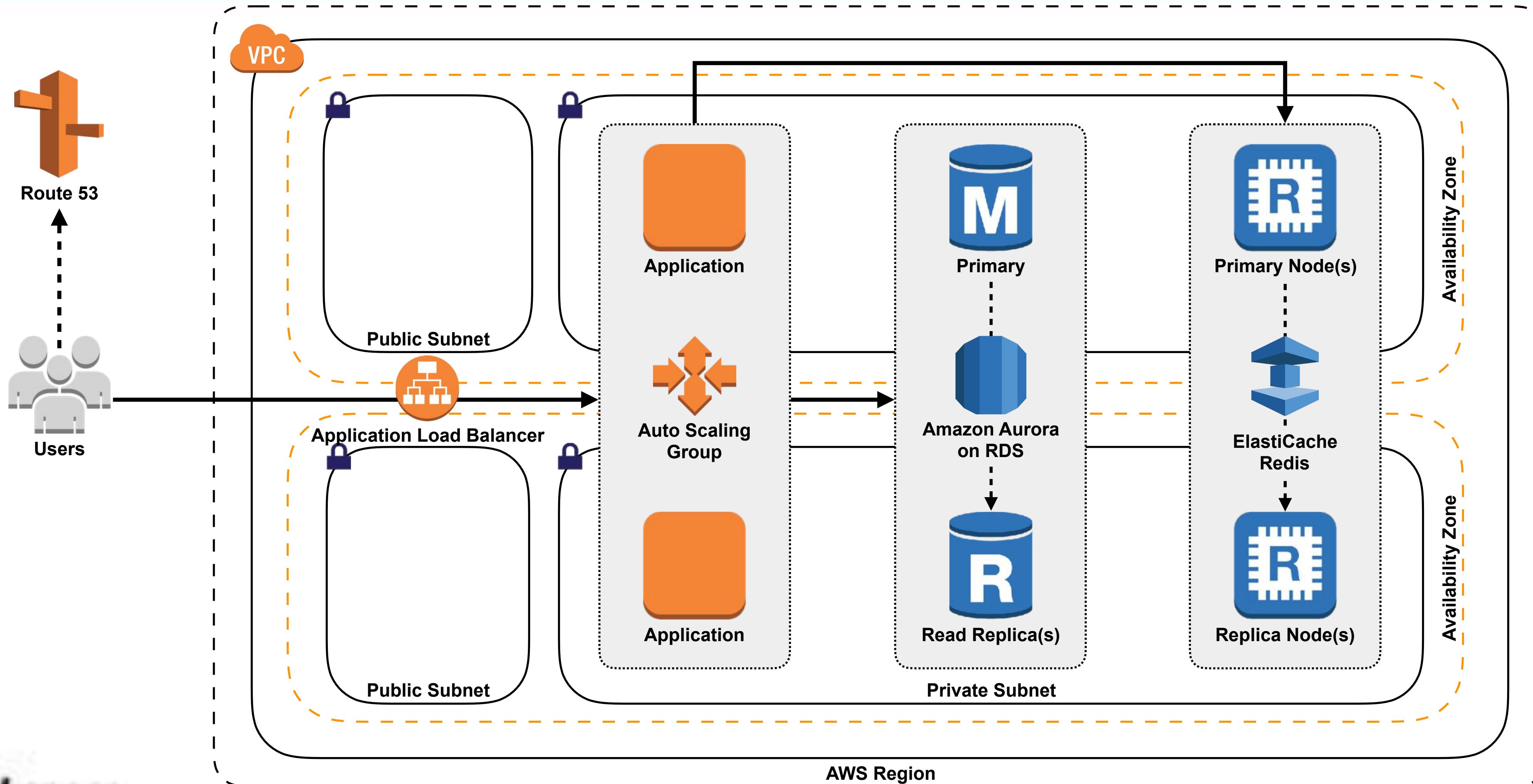
- In-memory cache as a service
- Automated setup and patching
- Memcached or Redis
- Fast!
- Easily scalable



Amazon ElastiCache



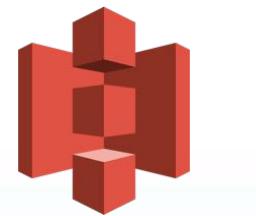
Amazon ElastiCache



Simple Storage Service (S3)



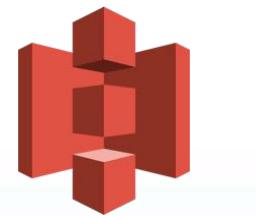
Simple Storage Service (S3)



- Pay-as-you-go object storage



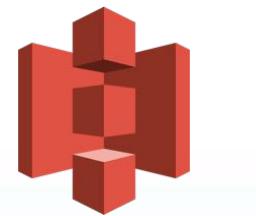
Simple Storage Service (S3)



- Pay-as-you-go object storage
- 5 TB per bucket with files up to 5 GB



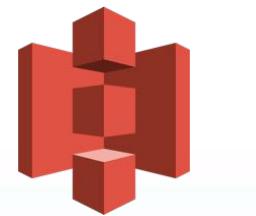
Simple Storage Service (S3)



- Pay-as-you-go object storage
- 5 TB per bucket with files up to 5 GB
- 99.999999999% (11 nines) durability



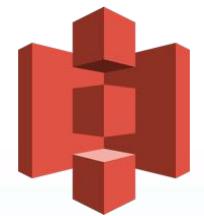
Simple Storage Service (S3)



- Pay-as-you-go object storage
- 5 TB per bucket with files up to 5 GB
- 99.99999999% (11 nines) durability
- File versioning



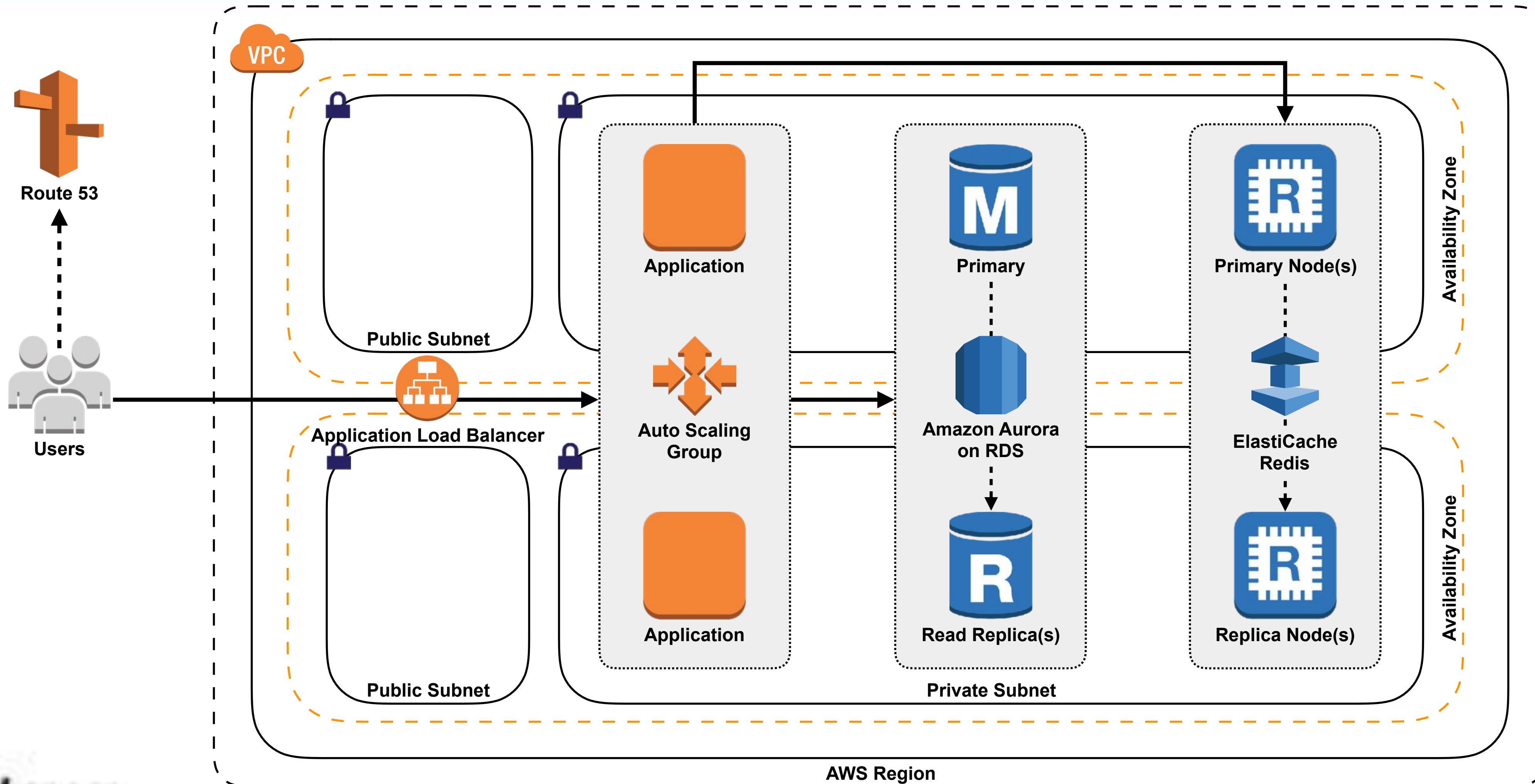
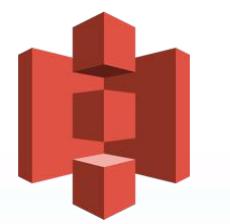
Simple Storage Service (S3)



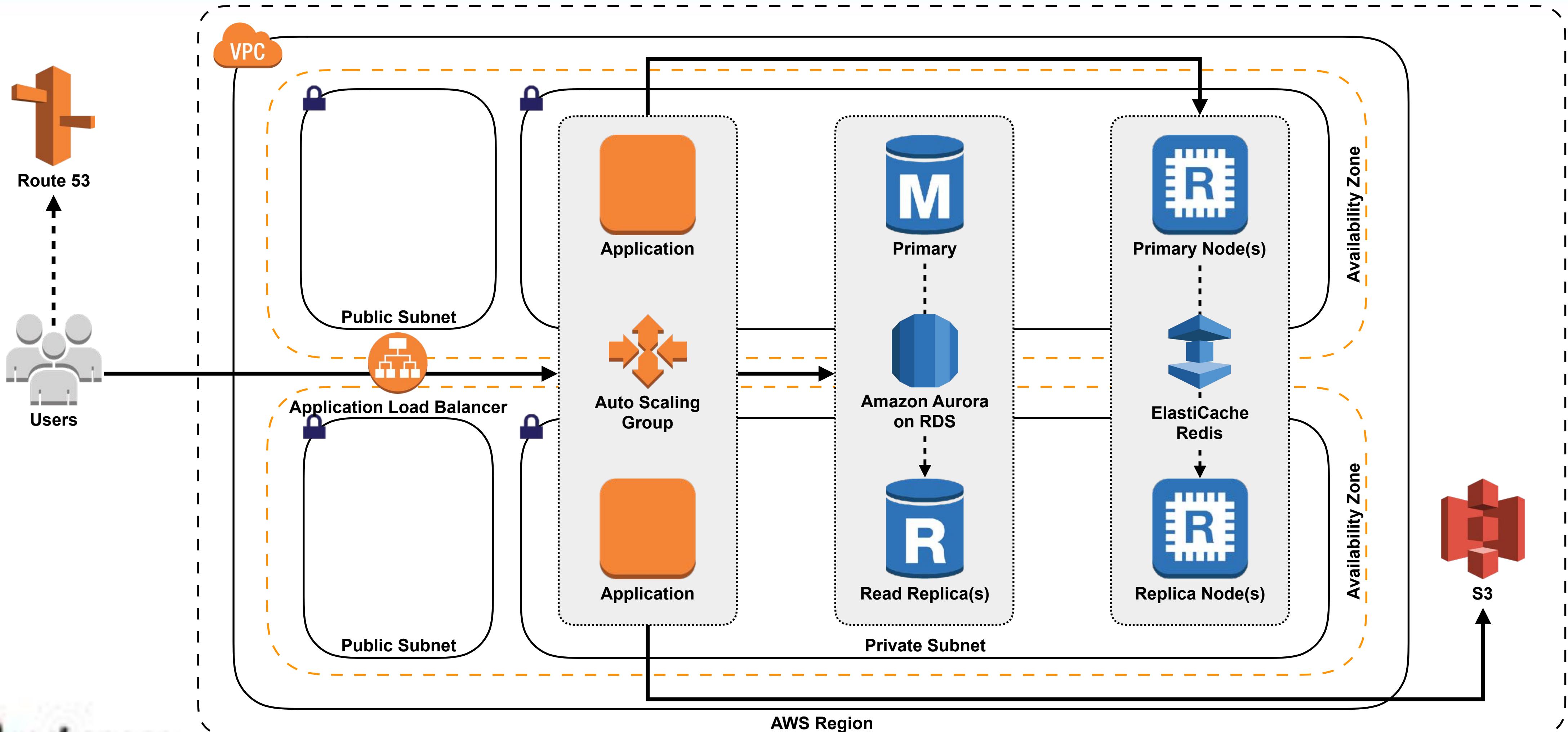
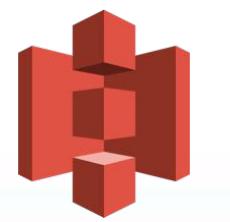
- Pay-as-you-go object storage
- 5 TB per bucket with files up to 5 GB
- 99.99999999% (11 nines) durability
- File versioning
- Static website hosting



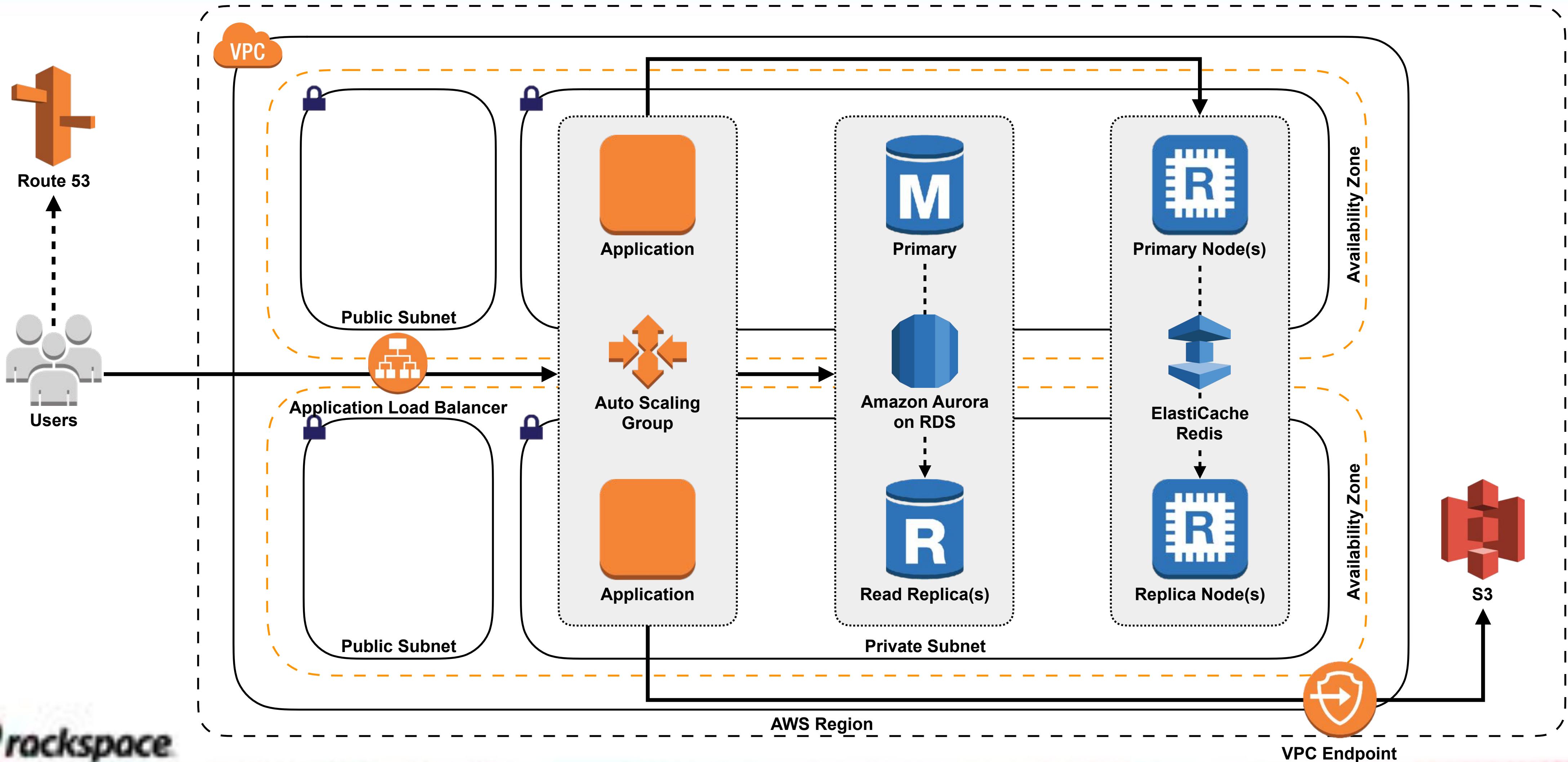
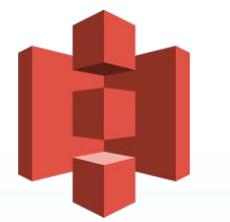
Simple Storage Service (S3)



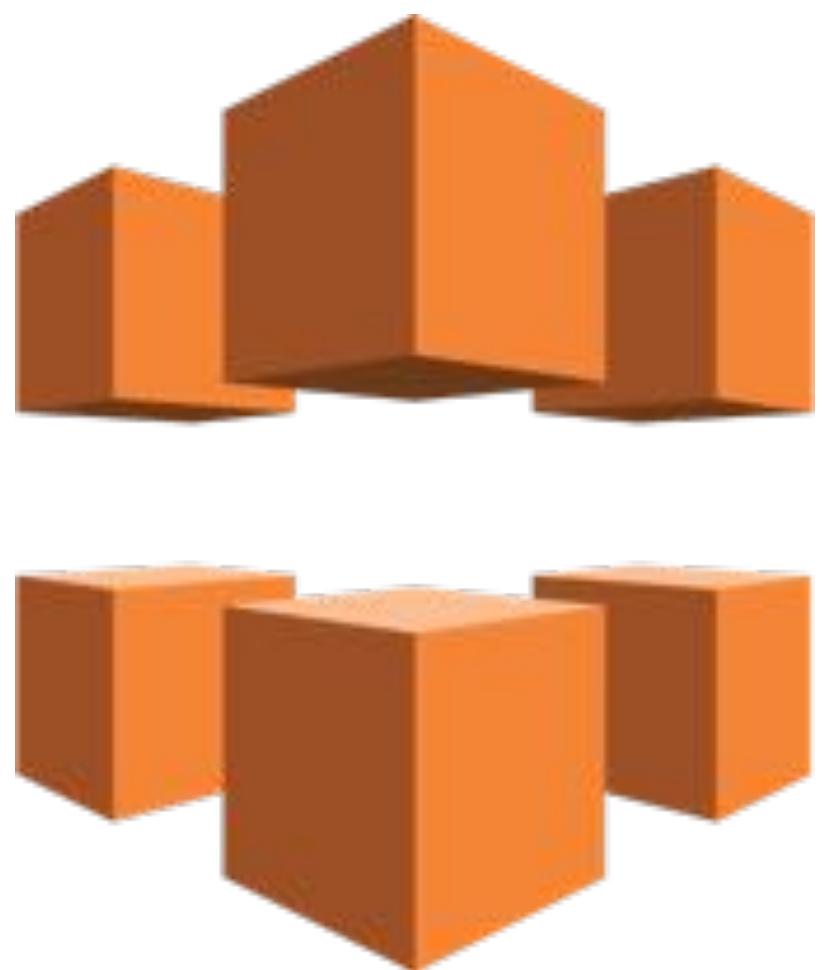
Simple Storage Service (S3)



Simple Storage Service (S3)

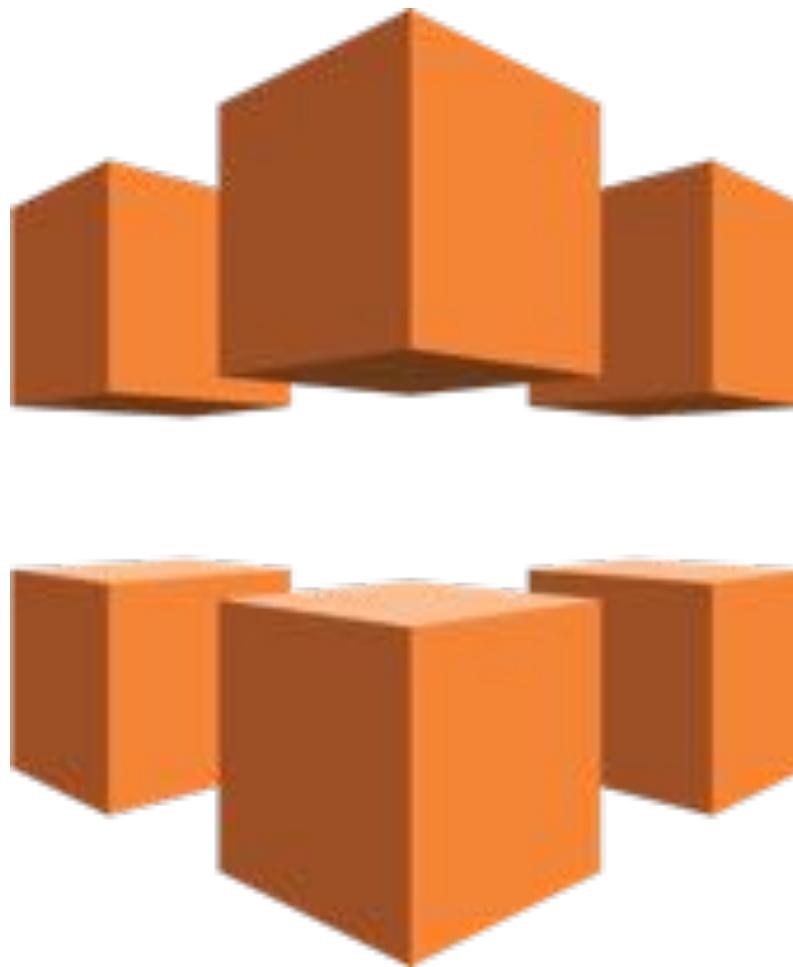


Amazon CloudFront

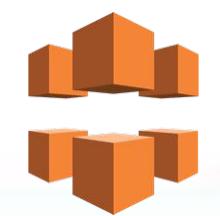


Amazon CloudFront

- Global content delivery network (CDN)

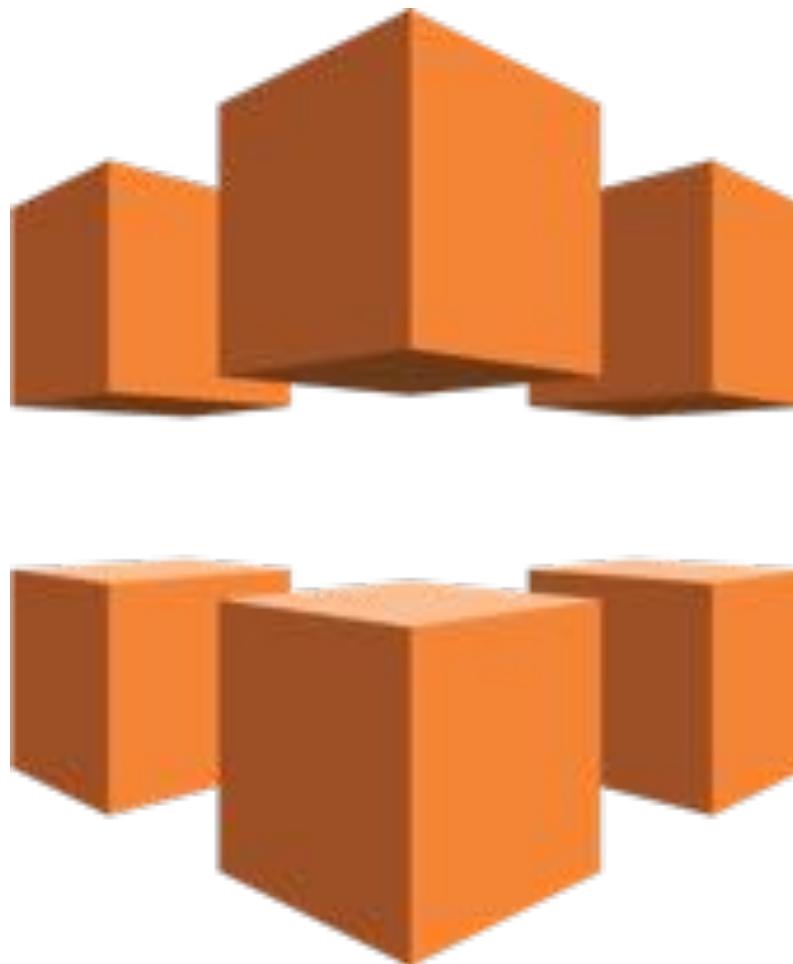


100+ Edge Locations across 24 countries



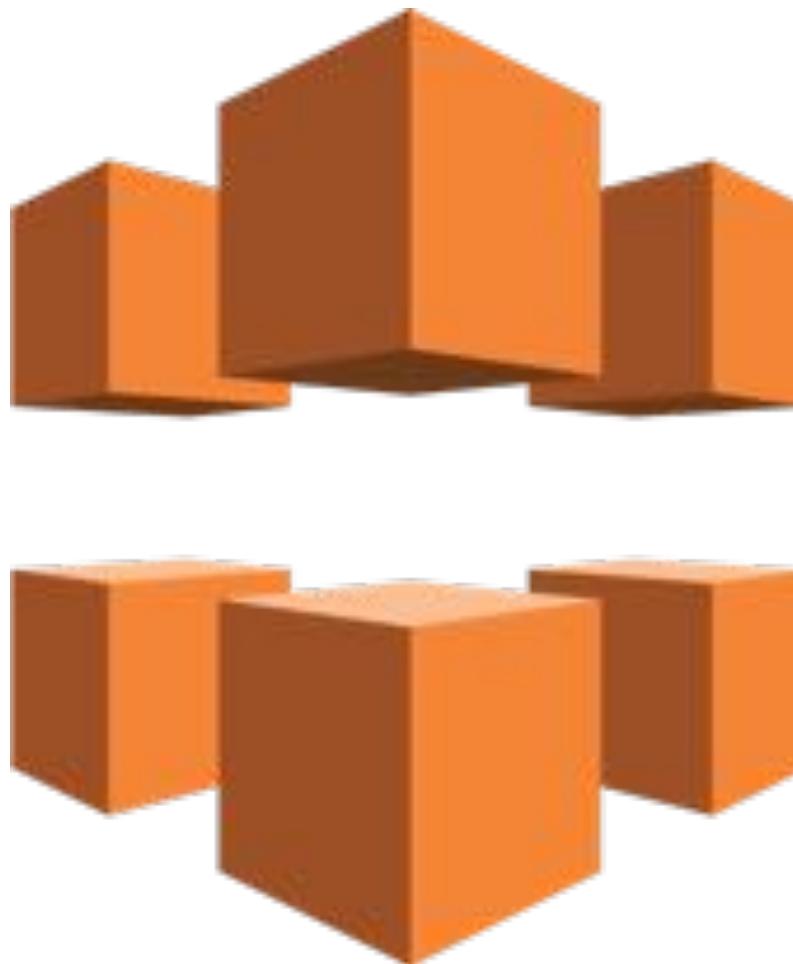
Amazon CloudFront

- Global content delivery network (CDN)
- Deep integration with EC2, ELB, and S3



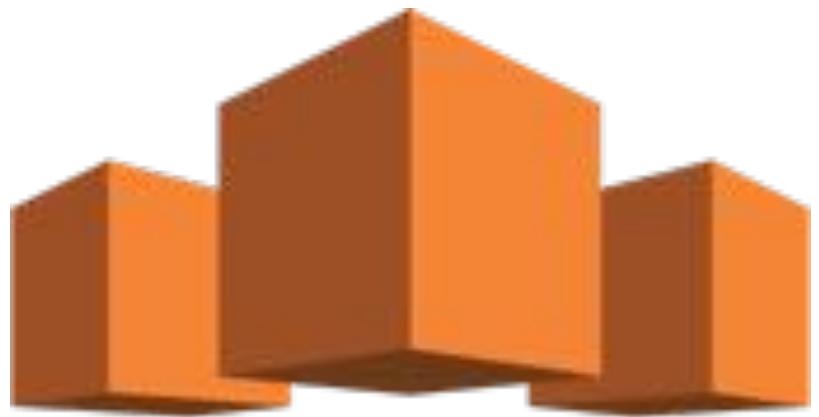
Amazon CloudFront

- Global content delivery network (CDN)
- Deep integration with EC2, ELB, and S3
- SSL/TLS available

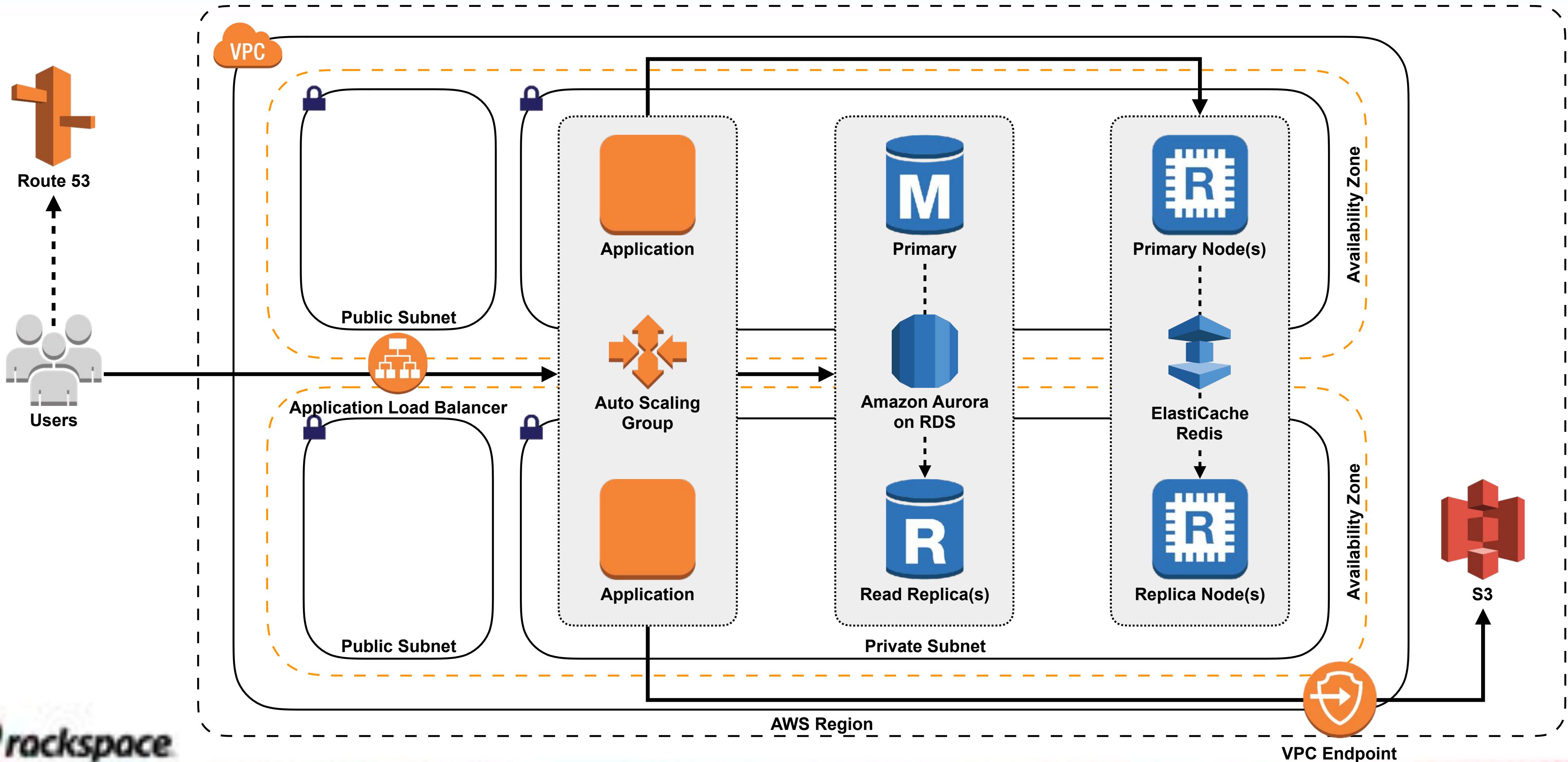
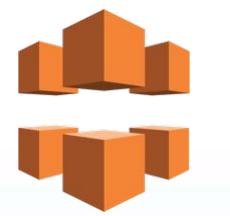


Amazon CloudFront

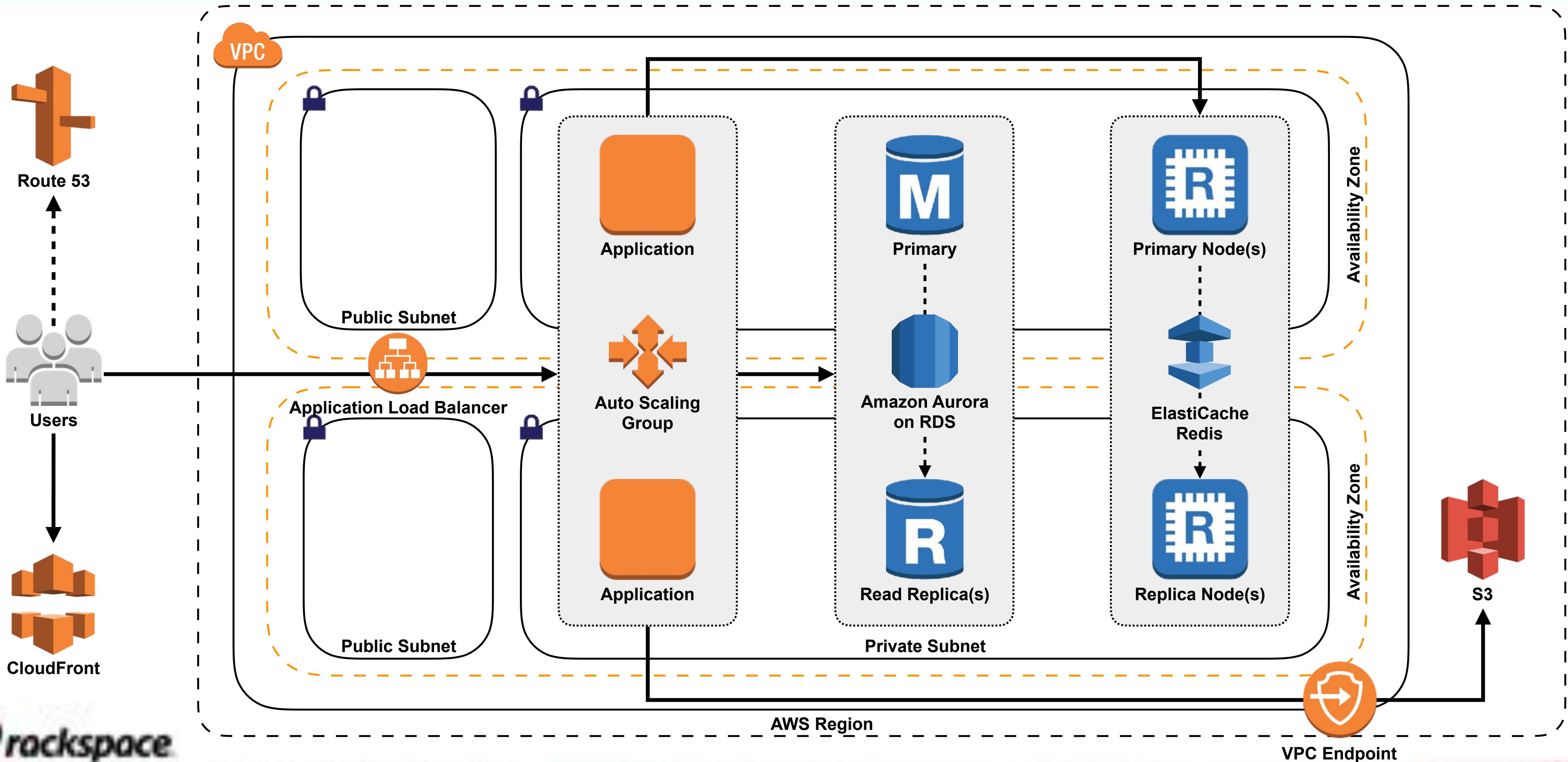
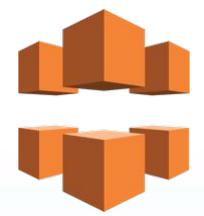
- Global content delivery network (CDN)
- Deep integration with EC2, ELB, and S3
- SSL/TLS available
- Private content support



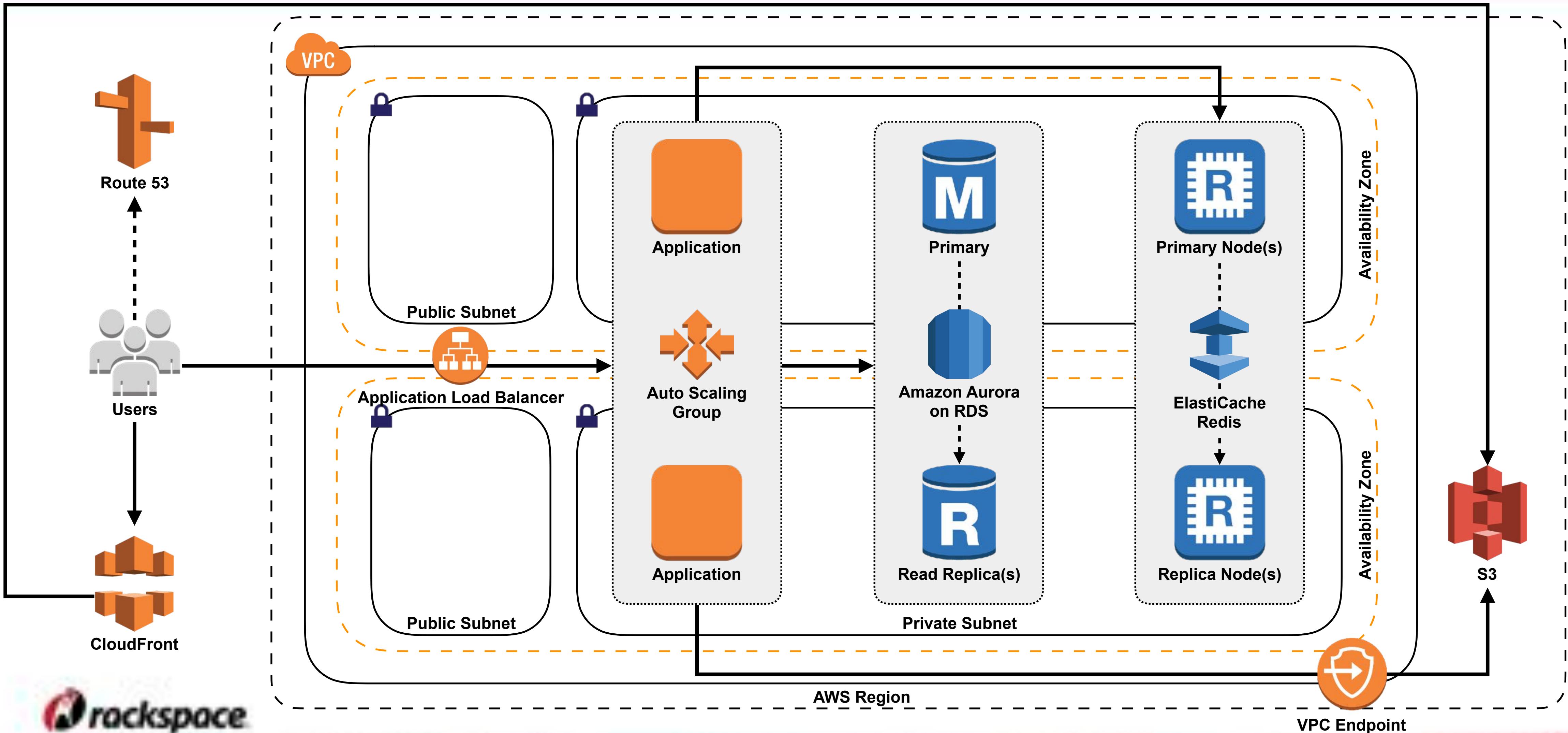
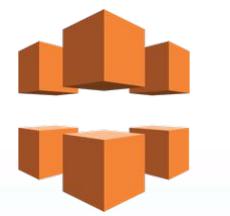
Amazon CloudFront



Amazon CloudFront



Amazon CloudFront



<https://join.d.in/talk/97b36>

Thanks!

<https://joind.in/talk/97b36>