Distributed Systems

Lesson 10

University of New York in Tirana
Master of Science in Computer Science
Prof. Dr. Marenglen Biba

Lesson 10

01: Introduction

02: Architectures

03: Processes

04: Communication

05: Naming

06: Synchronization

07: Consistency & Replication

08: Fault Tolerance

09: Security

10: Distributed Object-Based Systems

11: Distributed File Systems

12: Distributed Web-Based Systems

13: Distributed Coordination-Based Systems

14: Amazon Web Services

PART I

01: Introduction
02: Architectures
03: Processes
04: Communication
05: Naming
06: Synchronization
07: Consistency & Replication
08: Fault Tolerance
09: Security
10: Distributed Object-Based Systems
11: Distributed File Systems
12: Distributed Web-Based Systems
13: Distributed Coordination-Based Systems

Overall Approach

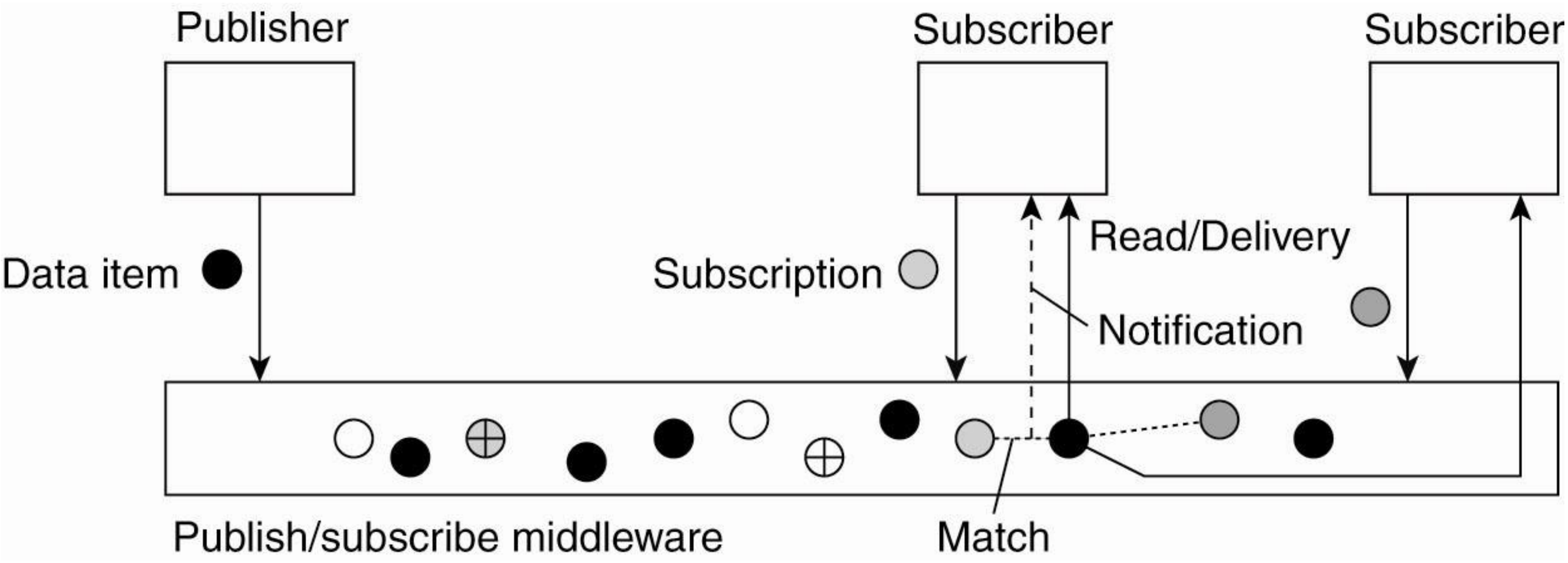


Figure 13-2. The principle of exchanging data items between publishers and subscribers.

Publish/subscribe systems in action

- Example: Reading stocks
  - https://www.tibco.com
  - https://demos.lightstreamer.com/

Java Spaces

- A tuple space is an implementation of the associative memory paradigm for parallel/distributed computing.
- It provides a repository of tuples that can be accessed concurrently.
- The implementation of tuple spaces for Java is JavaSpaces.
- JavaSpaces is a service specification providing a distributed object exchange and coordination mechanism for Java objects.
- It is used to store the distributed system state and implement distributed algorithms.
- In a JavaSpace, all communication partners (peers) communicate and coordinate by sharing state.

Example: Jini and JavaSpaces

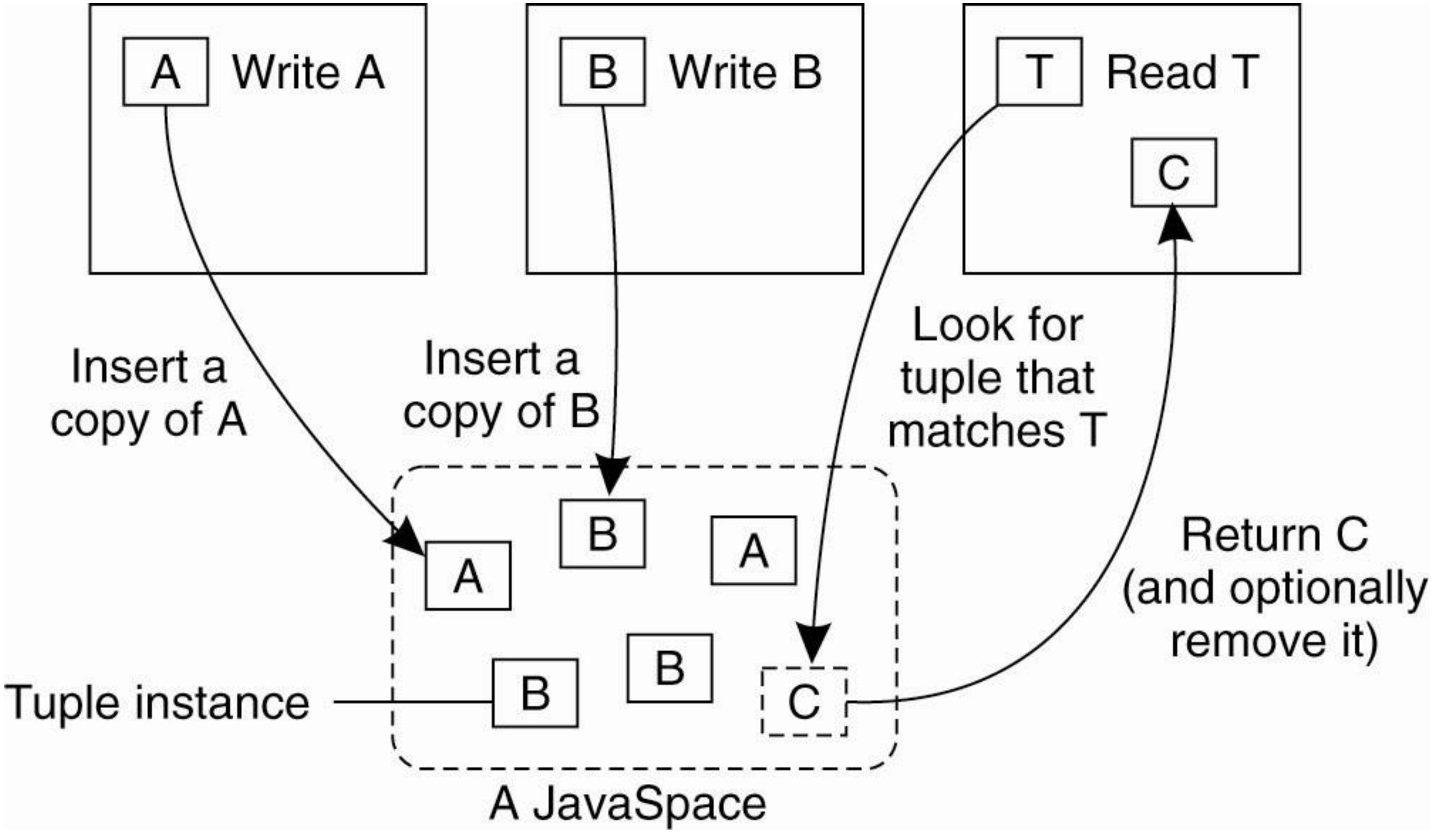
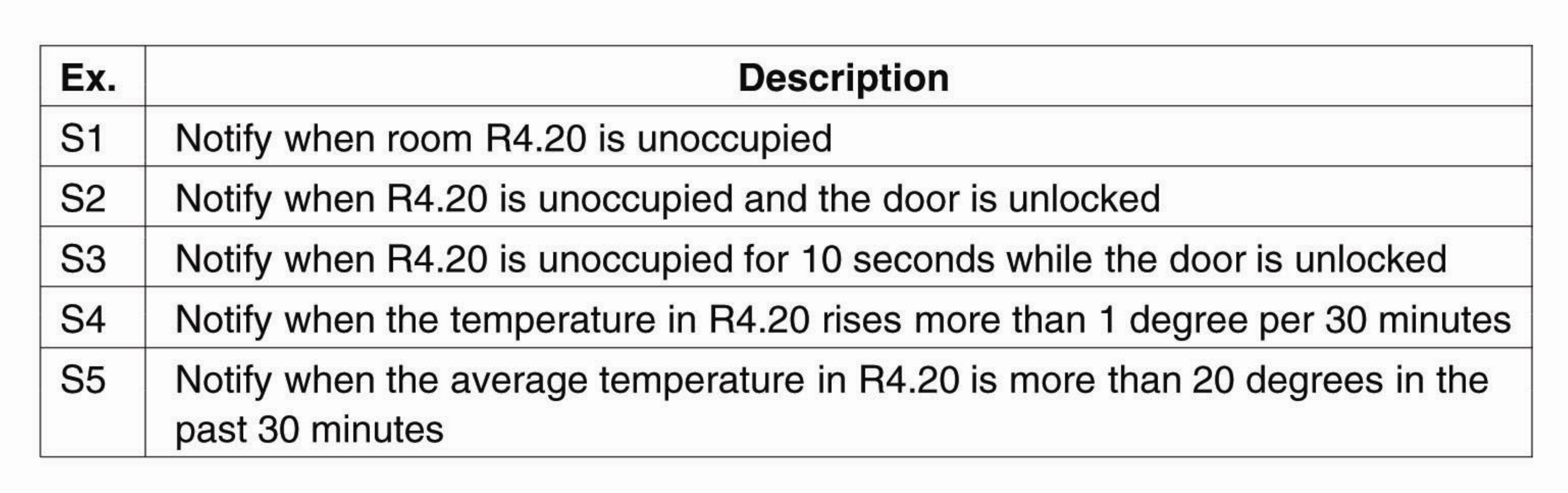


Figure 13-3. The general organization of a JavaSpace in Jini.

Events

- Events complicate the processing of subscriptions.
- Consider the subscription:
  - “Notify when room R4.20 is unoccupied and the door is unlocked”.
  - A distributed system supporting these subscriptions could be implemented by placing sensors.
- We would need to compose such primitive events into a publishable data item to which processes can then subscribe.
  - Event composition is hard!

Describing Composite Events (1)



Ex.	Description
S1	Notify when room R4.20 is unoccupied
S2	Notify when R4.20 is unoccupied and the door is unlocked
S3	Notify when R4.20 is unoccupied for 10 seconds while the door is unlocked
S4	Notify when the temperature in R4.20 rises more than 1 degree per 30 minutes
S5	Notify when the average temperature in R4.20 is more than 20 degrees in the past 30 minutes

Figure 13-9. Examples of events in a distributed system.

Describing Composite Events (2)

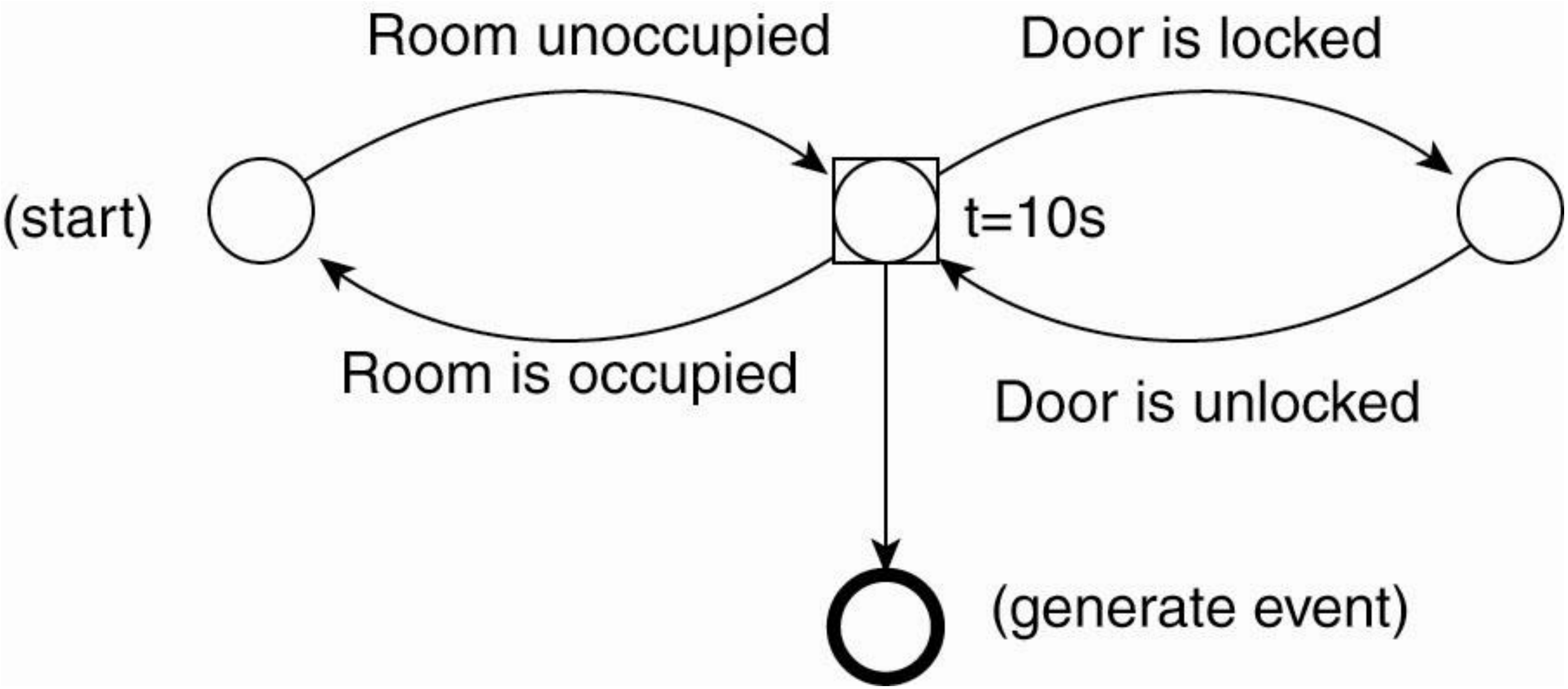


Figure 13-10. The finite state machine for subscription S3

Describing Composite Events (3)

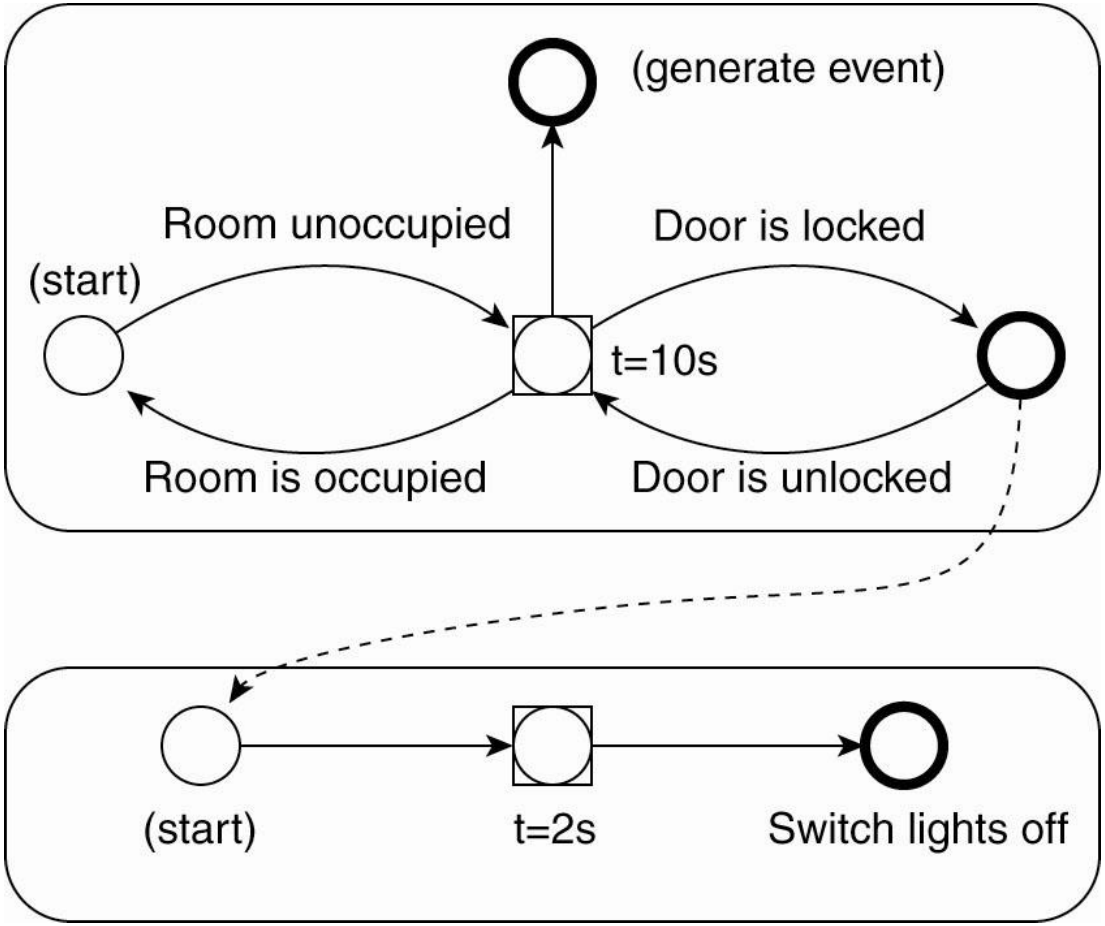


Figure 13-11. Two coupled FSMs.

Security in publish/subscribe systems

- Information confidentiality
  - It is sometimes important to disallow the middleware to inspect published data.
    - Solved through end-to-end encryption
- Subscription confidentiality
  - Subscriptions may not be disclosed to the middleware
- Publication confidentiality
  - Publishers may want to explicitly restrict the group of possible subscribers.

Decoupling Publishers from Subscribers

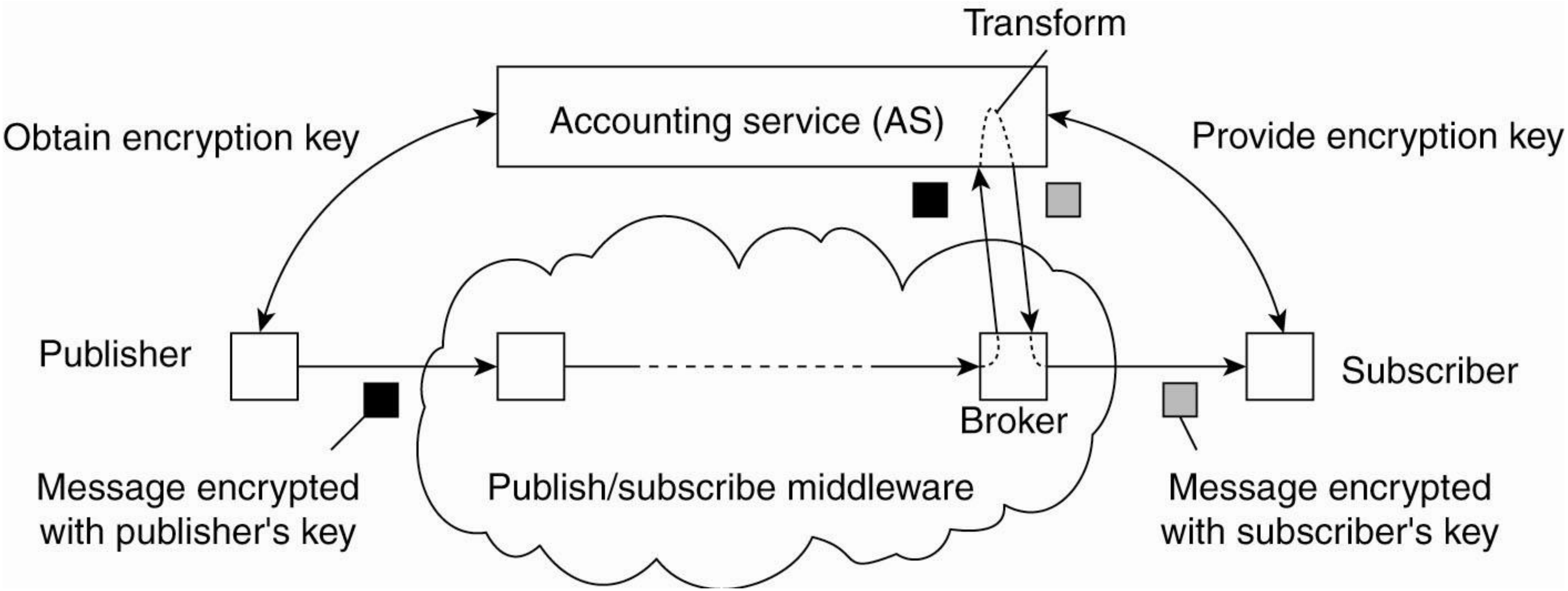

Transform
Obtain encryption key
Accounting service (AS)
Provide encryption key
Publisher
Subscriber
Broker
Message encrypted with publisher's key
Publish/subscribe middleware
Message encrypted with subscriber's key

Figure 13-18. Decoupling publishers from subscribers using an additional trusted service.

End of PART I

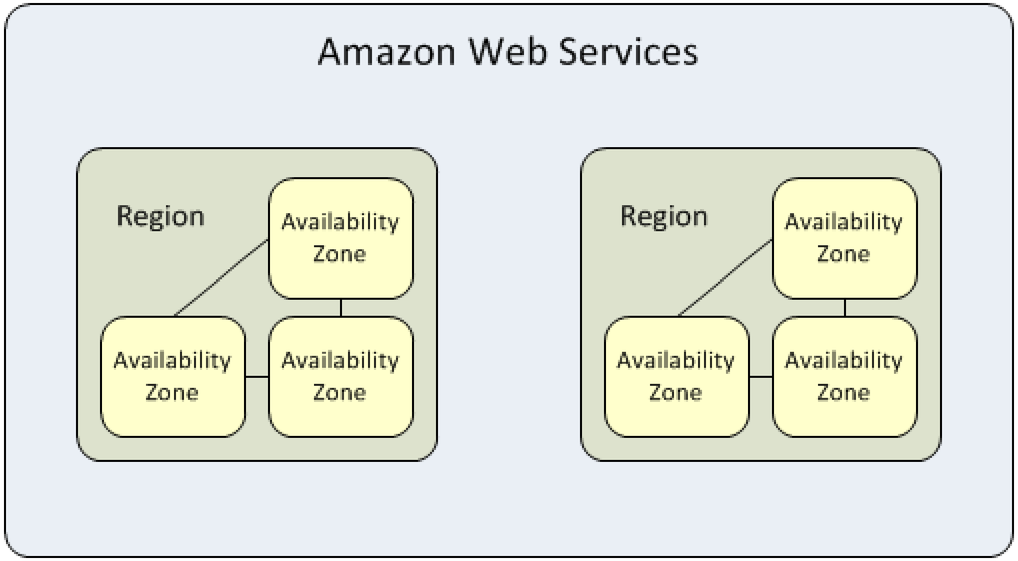
- Readings
  - – Distributed Systems, Chapter 13

PART II

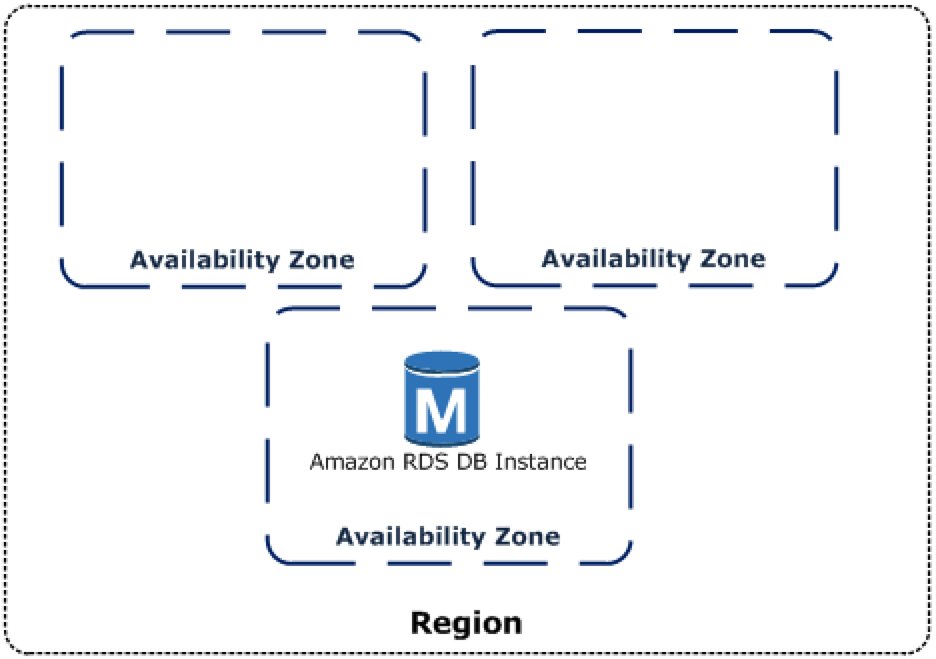
01: Introduction
02: Architectures
03: Processes
04: Communication
05: Naming
06: Synchronization
07: Consistency & Replication
08: Fault Tolerance
09: Security
10: Distributed Object-Based Systems
11: Distributed File Systems
12: Distributed Web-Based Systems
13: Distributed Coordination-Based Systems
14: Amazon Web Services

Amazon Web Services
Cloud Computing Services

Amazon WS: Regions and AZ



Amazon WS

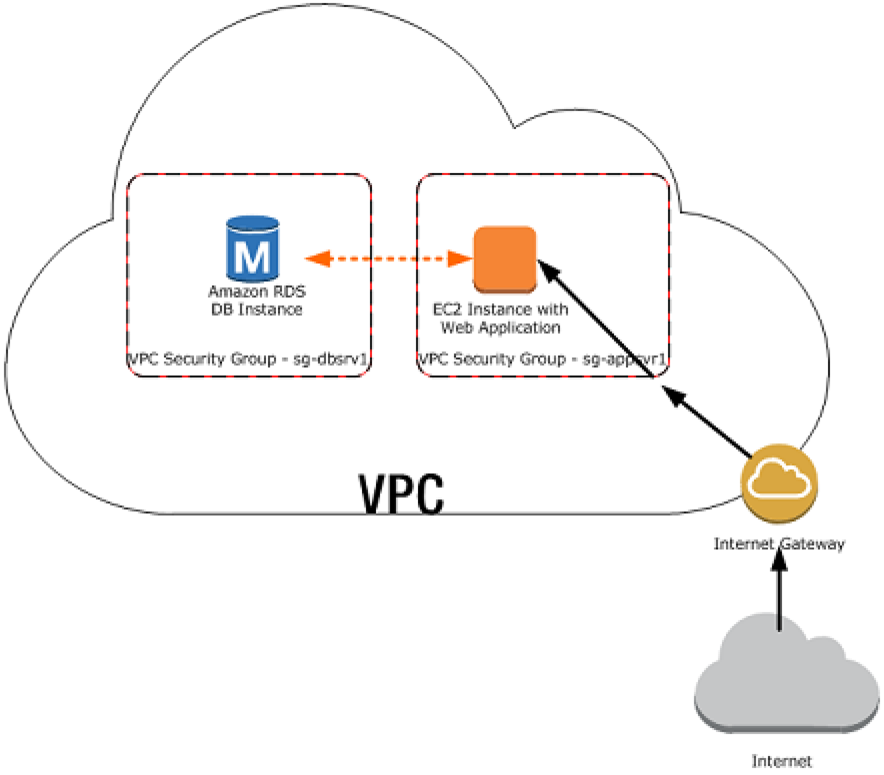

Availability Zone
Availability Zone
Amazon RDS DB Instance
Availability Zone
Region

Regions in AWS

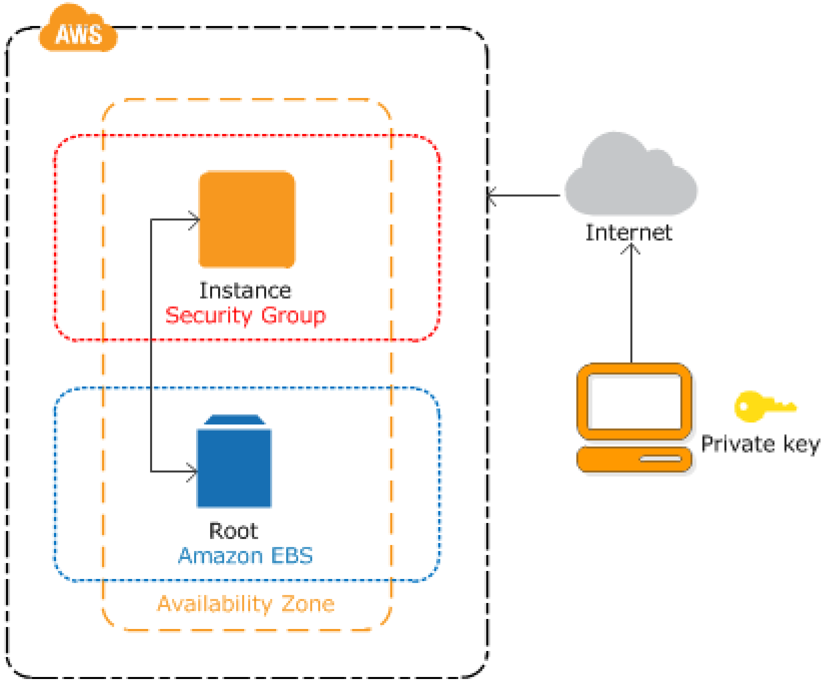
Region Name	Region	Endpoint	Protocol
US East (Ohio)	us-east-2	rds.us-east-2.amazonaws.com	HTTPS
US East (N. Virginia)	us-east-1	rds.us-east-1.amazonaws.com	HTTPS
US West (N. California)	us-west-1	rds.us-west-1.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	rds.us-west-2.amazonaws.com	HTTPS
Asia Pacific (Hong Kong)	ap-east-1	rds.ap-east-1.amazonaws.com	HTTPS
Asia Pacific (Mumbai)	ap-south-1	rds.ap-south-1.amazonaws.com	HTTPS
Asia Pacific (Osaka-Local)	ap-northeast-3	rds.ap-northeast-3.amazonaws.com	HTTPS
Asia Pacific (Seoul)	ap-northeast-2	rds.ap-northeast-2.amazonaws.com	HTTPS
Asia Pacific (Singapore)	ap-southeast-1	rds.ap-southeast-1.amazonaws.com	HTTPS
Asia Pacific (Sydney)	ap-southeast-2	rds.ap-southeast-2.amazonaws.com	HTTPS
Asia Pacific (Tokyo)	ap-northeast-1	rds.ap-northeast-1.amazonaws.com	HTTPS
Canada (Central)	ca-central-1	rds.ca-central-1.amazonaws.com	HTTPS
China (Beijing)	cn-north-1	rds.cn-north-1.amazonaws.com.cn	HTTPS
China (Ningxia)	cn-northwest-1	rds.cn-	HTTPS

EU (Frankfurt)	eu-central-1	rds.eu-central-1.amazonaws.com	HTTPS
EU (Ireland)	eu-west-1	rds.eu-west-1.amazonaws.com	HTTPS
EU (London)	eu-west-2	rds.eu-west-2.amazonaws.com	HTTPS
EU (Paris)	eu-west-3	rds.eu-west-3.amazonaws.com	HTTPS
EU (Stockholm)	eu-north-1	rds.eu-north-1.amazonaws.com	HTTPS
South America (São Paulo)	sa-east-1	rds.sa-east-1.amazonaws.com	HTTPS
AWS GovCloud (US-East)	us-gov-east-1	rds.us-gov-east-1.amazonaws.com	HTTPS
AWS GovCloud (US)	us-gov-west-1	rds.us-gov-west-1.amazonaws.com	HTTPS

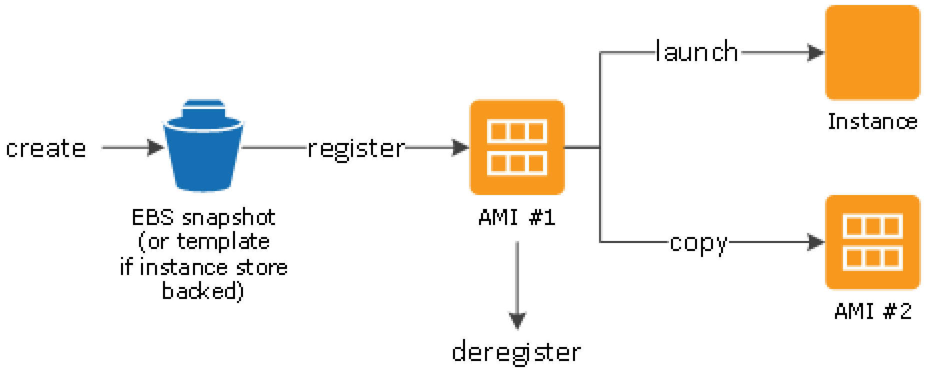
Amazon VPC


Amazon RDS
DB Instance
VPC Security Group - sg-dbsrv1
EC2 Instance with
Web Application
VPC Security Group - sg-appsvr1
VPC
Internet Gateway
Internet

Amazon Elastic Compute Cloud (Amazon EC2)



Amazon AMI


create
register
launch
copy
Instance
EBS snapshot
(or template
if instance store
backed)
AMI #1
AMI #2
deregister

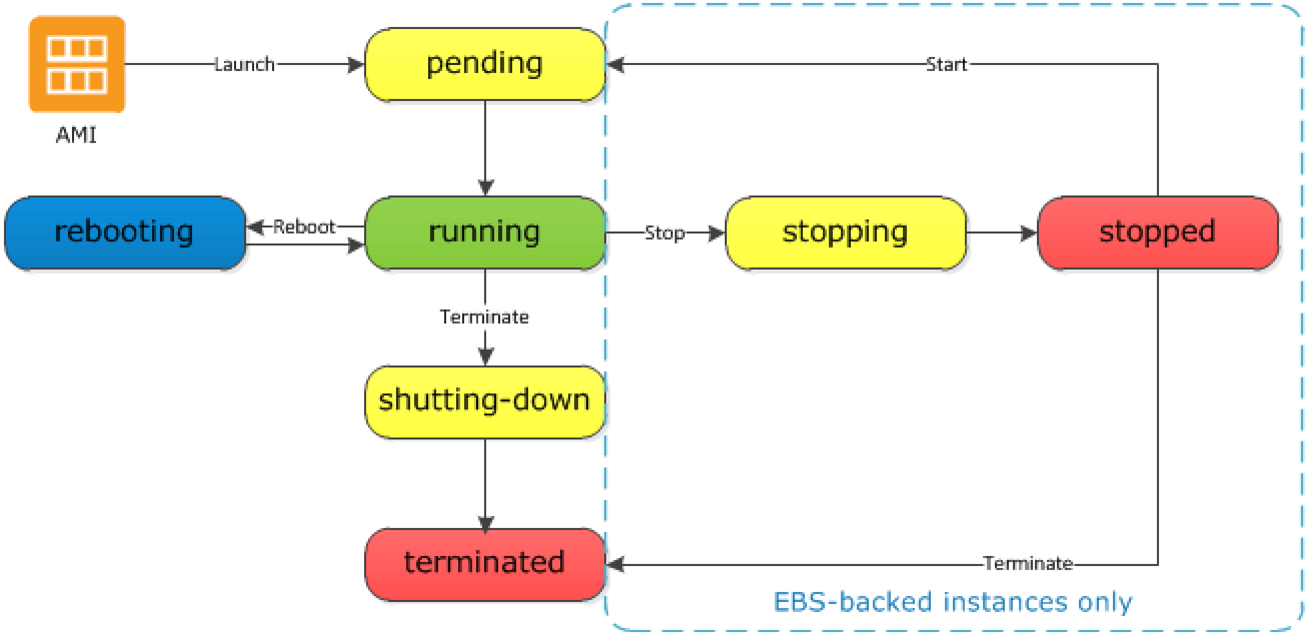
Instances

Instance Name	vCPUs	RAM	Local Storage	EBS-Optimized Bandwidth	Network Bandwidth
m5ad.large	2	8 GiB	1 x 75 GB NVMe SSD	Up to 2.120 Gbps	Up to 10 Gbps
m5ad.xlarge	4	16 GiB	1 x 150 GB NVMe SSD	Up to 2.120 Gbps	Up to 10 Gbps
m5ad.2xlarge	8	32 GiB	1 x 300 GB NVMe SSD	Up to 2.120 Gbps	Up to 10 Gbps
m5ad.4xlarge	16	64 GiB	2 x 300 GB NVMe SSD	2.120 Gbps	Up to 10 Gbps
m5ad.12xlarge	48	192 GiB	2 x 900 GB NVMe SSD	5 Gbps	10 Gbps
m5ad.24xlarge	96	384 GiB	4 x 900 GB NVMe SSD	10 Gbps	20 Gbps

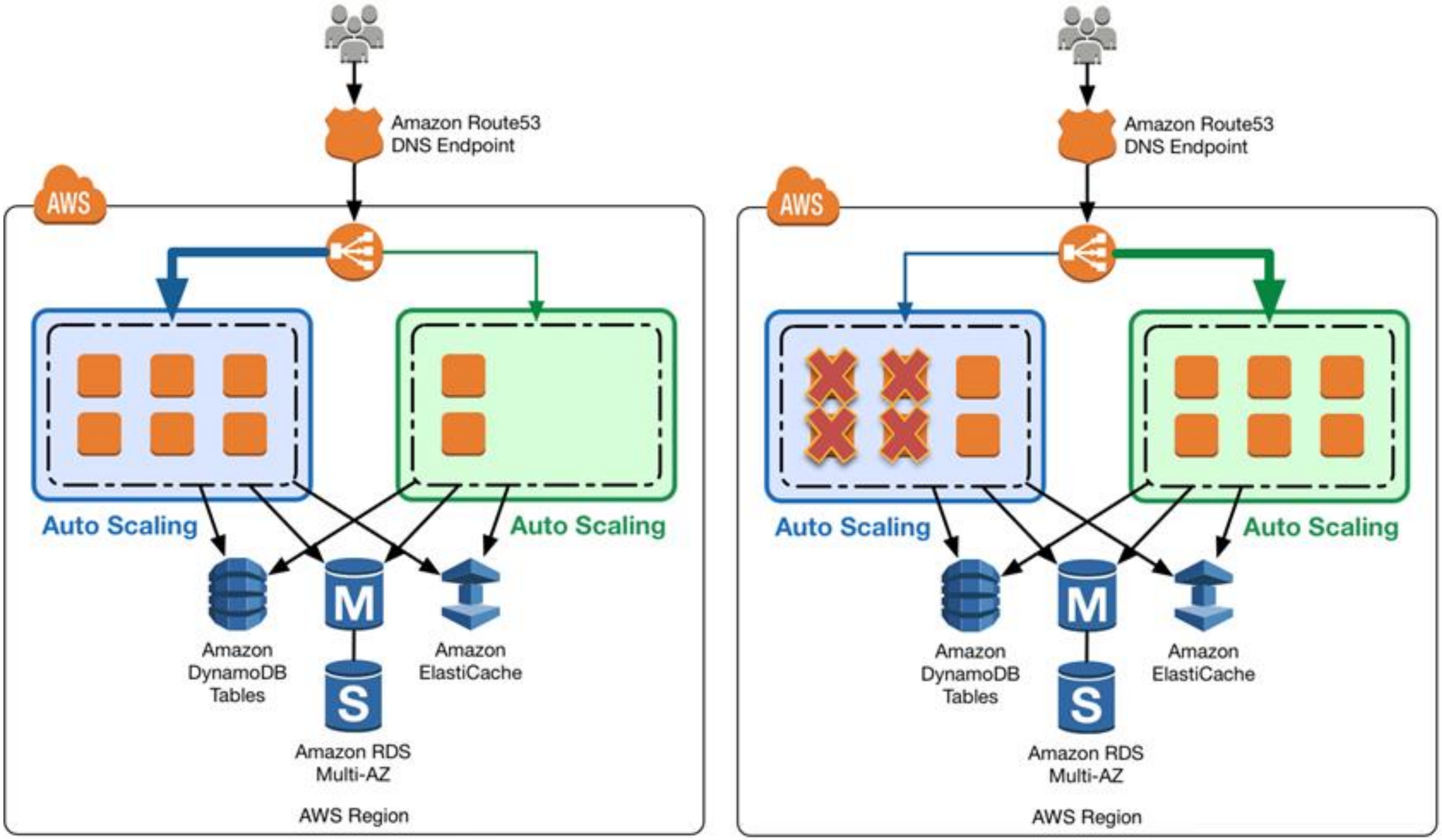
Instances

Instance Name	vCPUs	RAM	Local Storage	EBS-Optimized Bandwidth	Network Bandwidth
r5ad.large	2	16 GiB	1 x 75 GB NVMe SSD	Up to 2.120 Gbps	Up to 10 Gbps
r5ad.xlarge	4	32 GiB	1 x 150 GB NVMe SSD	Up to 2.120 Gbps	Up to 10 Gbps
r5ad.2xlarge	8	64 GiB	1 x 300 GB NVMe SSD	Up to 2.120 Gbps	Up to 10 Gbps
r5ad.4xlarge	16	128 GiB	2 x 300 GB NVMe SSD	2.120 Gbps	Up to 10 Gbps
r5ad.12xlarge	48	384 GiB	2 x 900 GB NVMe SSD	5 Gbps	10 Gbps
r5ad.24xlarge	96	768 GiB	4 x 900 GB NVMe SSD	10 Gbps	20 Gbps

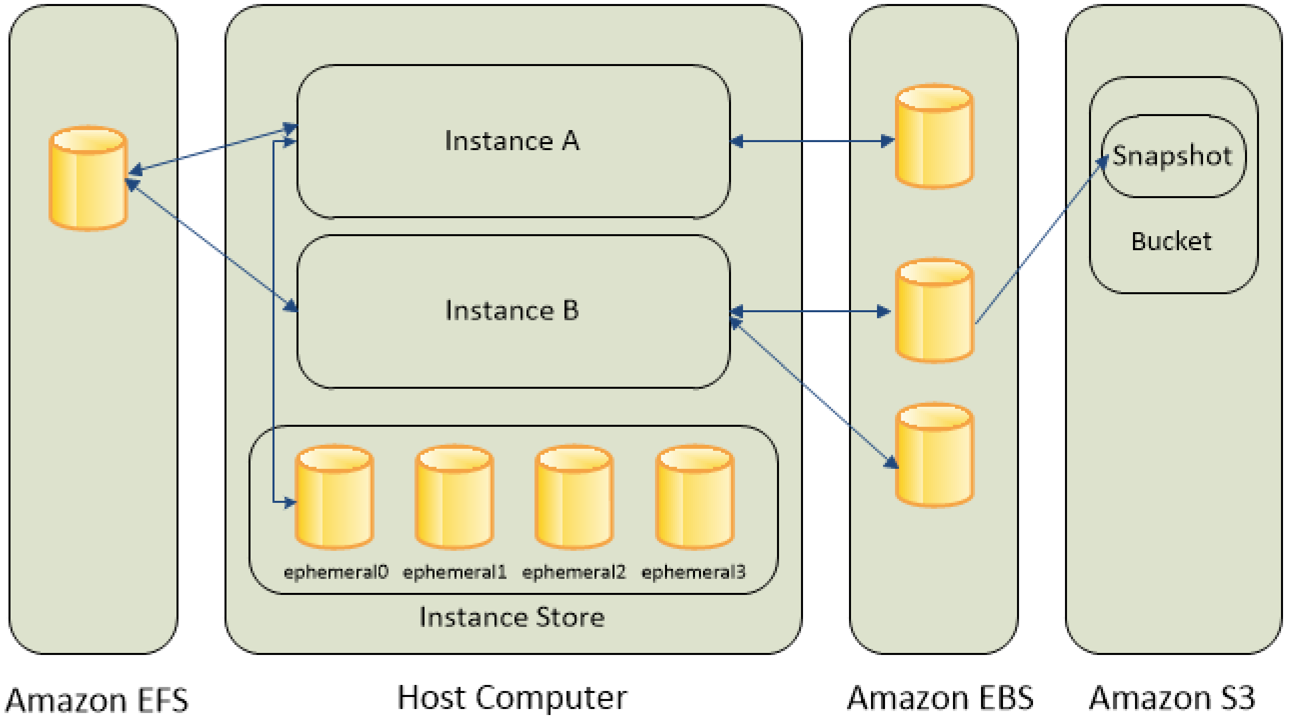
Instances


AMI
Launch
pending
Start
rebooting
Reboot
running
Stop
stopping
stopped
Terminate
shutting-down
terminated
Terminate
EBS-backed instances only

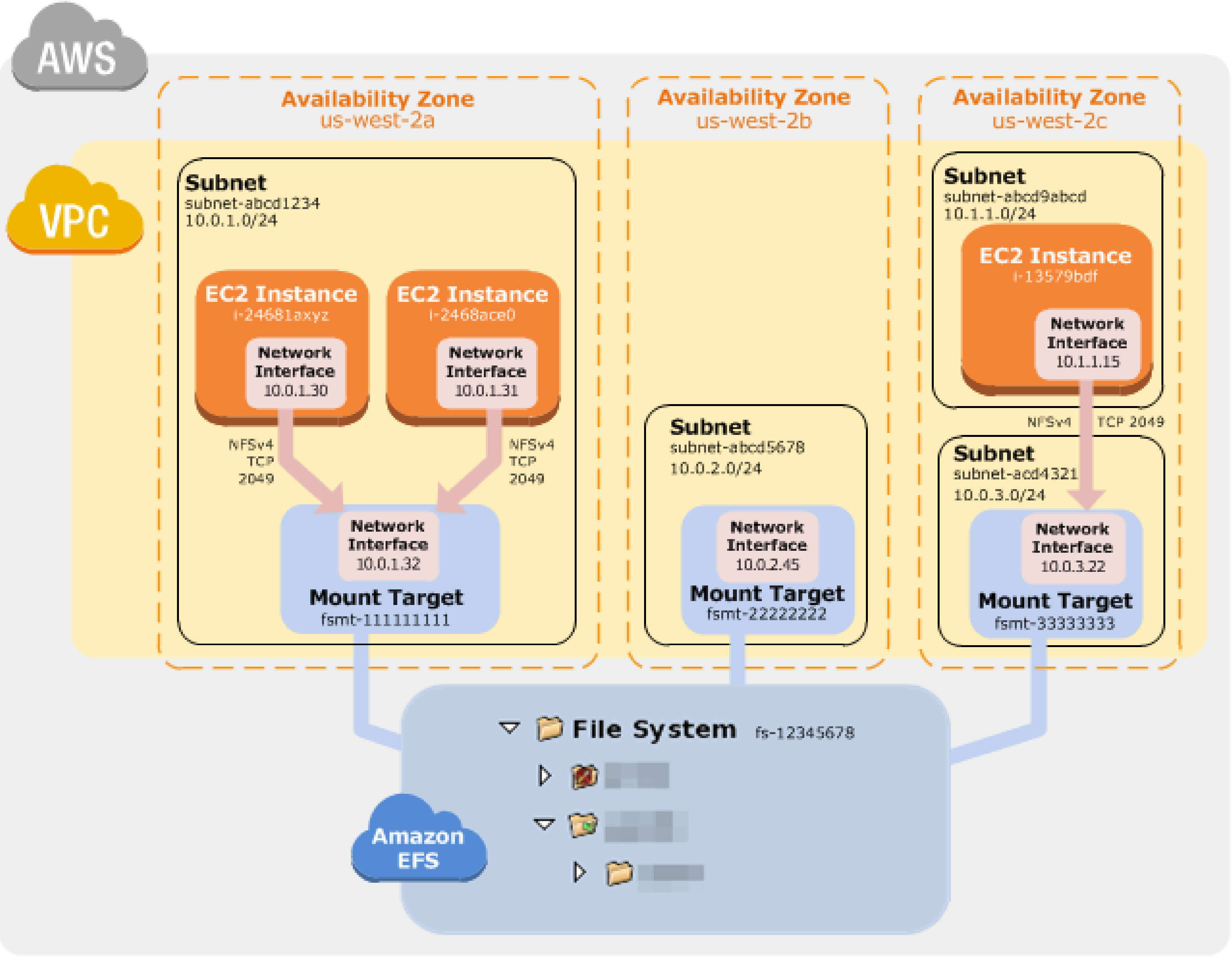
Auto scaling



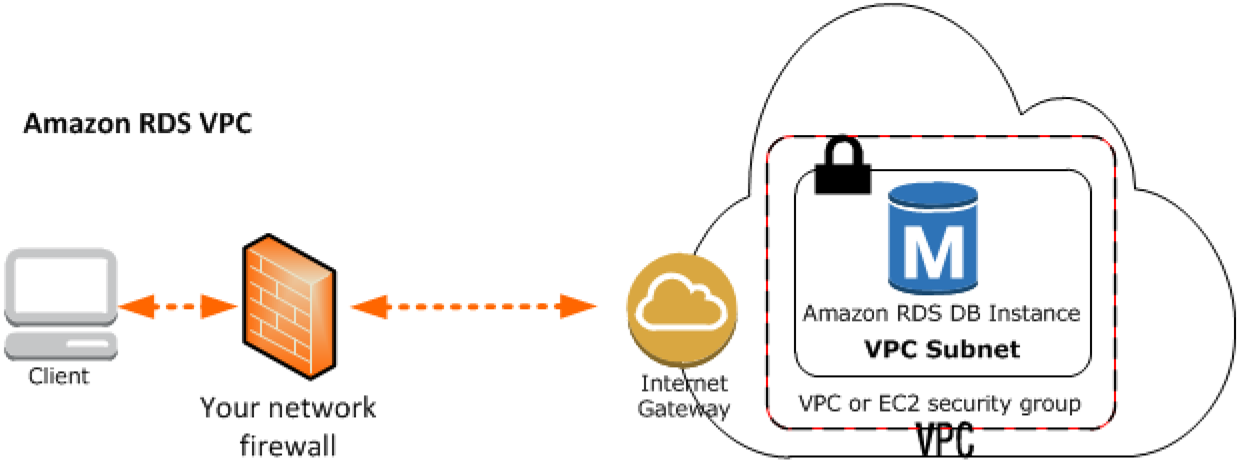
Amazon EBS


Instance A
Instance B
Snapshot
Bucket
ephemeral0
ephemeral1
ephemeral2
ephemeral3
Instance Store
Amazon EFS
Host Computer
Amazon EBS
Amazon S3

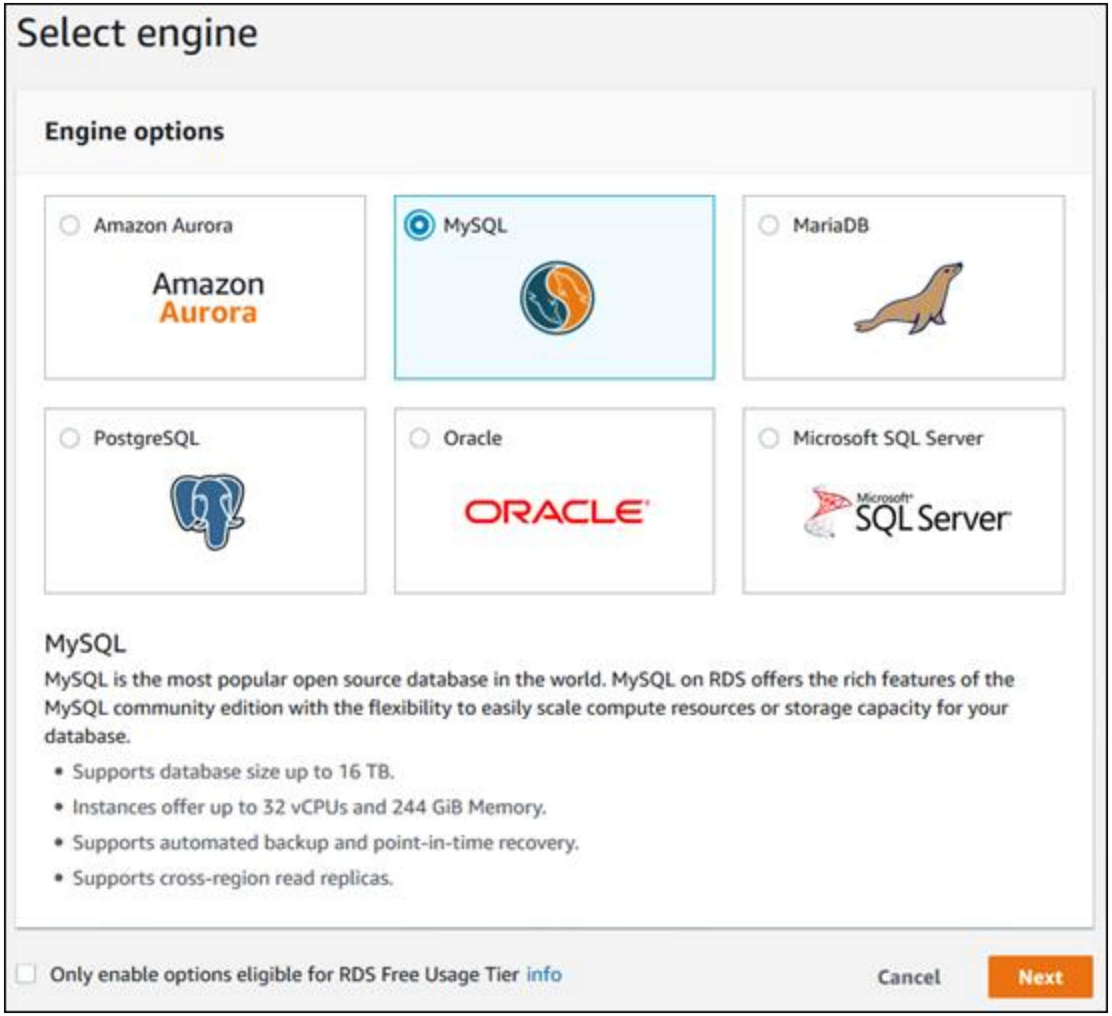
Amazon EFS


AWS
VPC
Availability Zone
us-west-2a
Availability Zone
us-west-2b
Availability Zone
us-west-2c
Subnet
subnet-abcd1234
10.0.1.0/24
EC2 Instance
i-24681axyz
Network Interface
10.0.1.30
EC2 Instance
i-2468ace0
Network Interface
10.0.1.31
NFSv4 TCP 2049
NFSv4 TCP 2049
Network Interface
10.0.1.32
Mount Target
fsmt-111111111
Subnet
subnet-abcd5678
10.0.2.0/24
Network Interface
10.0.2.45
Mount Target
fsmt-22222222
Subnet
subnet-abcd9abcd
10.1.1.0/24
EC2 Instance
i-13579bdf
Network Interface
10.1.1.15
NFSv4
TCP 2049
Subnet
subnet-acd4321
10.0.3.0/24
Network Interface
10.0.3.22
Mount Target
fsmt-33333333
File System fs-12345678
Amazon EFS

Amazon RDS


Amazon RDS VPC
Client
Your network firewall
Internet Gateway
Amazon RDS DB Instance
VPC Subnet
VPC or EC2 security group
VPC

DB Engines



Select engine

Engine options

○ Amazon Aurora

● MySQL

○ MariaDB

○ PostgreSQL

○ Oracle

○ Microsoft SQL Server

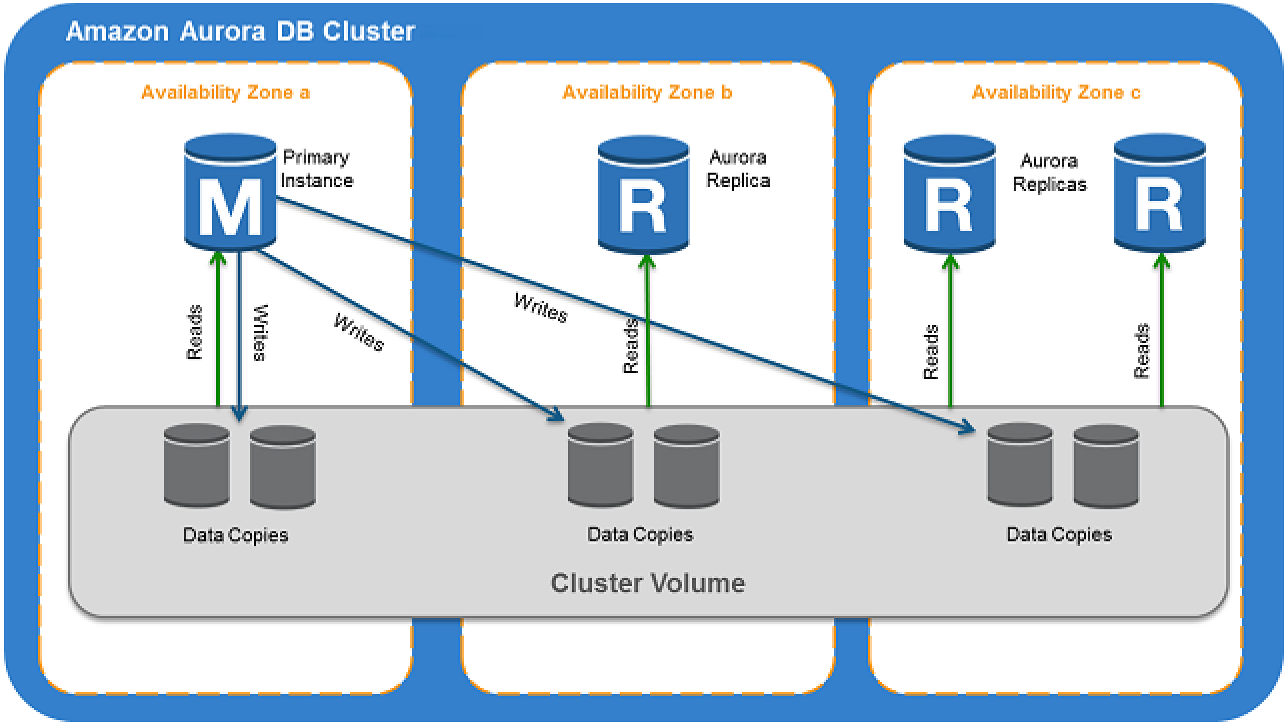
MySQL

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 16 TB.
- Instances offer up to 32 vCPUs and 244 GiB Memory.
- Supports automated backup and point-in-time recovery.
- Supports cross-region read replicas.

□ Only enable options eligible for RDS Free Usage Tier info

Amazon Aurora DB Clusters



End of Part II

- Reading
  - – AWS Documentation

Lab Session

- Web Server: Java EE 6 Web
- Web Interface Tier:
  - Servlets
- Processing Tier
  - Enteprise Java Beans
- Applications
  - Web Banking
- Follow the manual given in class.

End of Lab Session

- Readings
  - Lab Manual on EJBs + Servlets + MySQL + Client given in class