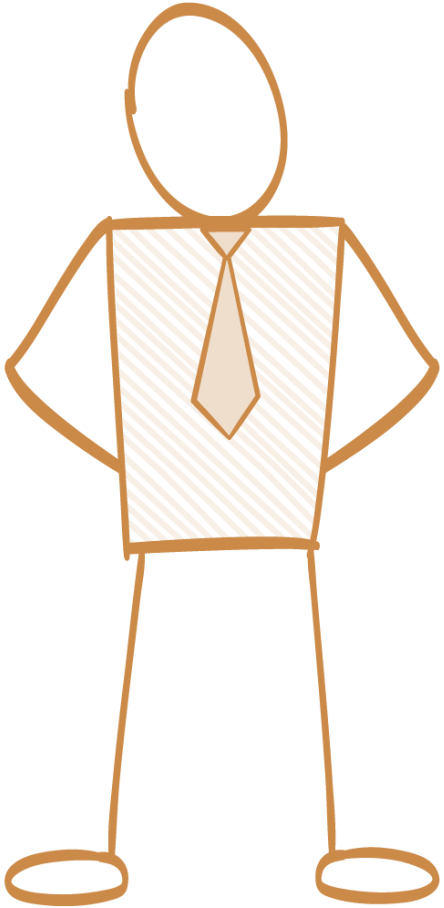


Module 13: Disaster Recovery and High Availability

amazon web services

Topics

- How High Availability and Disaster Recovery work together
- Building highly available systems on AWS
- Best practices for high availability and disaster recovery
- Common patterns of disaster recovery on AWS

amazon web services

Topics

- How High Availability and Disaster Recovery work together
- Building highly available systems on AWS
- Best practices for high availability and disaster recovery
- Common patterns of disaster recovery on AWS

amazon web services

How Availability and Disaster Recovery

- Think of it as a spectrum
- It's part of a business continuity plan
- It's not an all or nothing proposition
- In the face of internal or external events, how to you….
  - Keep your application running 24x7
  - Make sure your data is safe
  - Get an application back up after a major disaster

amazon web services

How Availability

- HA is one end of the spectrum
  - Rather than recovery with defined RTO/RPO, design for continuous availability
- Goal: application never goes down
  - Eliminate single points of failure
  - "Self Healing" app recovers automatically from component failures
  - Use graceful degradation if necessary
- Traditional IT model: HA is very expensive, suitable only for absolutely mission-critical apps

amazon web services

How Availability Terms

- Uptime: period when system is availability for use
- Downtime (or Outage): period when system is unavailable for normal function or offline
- Graceful Degradation: system continues to be available, but at a reduced level of service or function
- Availability: percentage uptime in a given period (nines)
  - –99,999% (“five nines”) availability= no more than about 5 minutes downtime per year

amazon web services

Disaster Recovery on the spectrum

- Recover from any event
- Recover Time Object (RTO)
  - Acceptable time period within which normal operation (or degraded operation) needs to be restored after event
- Recovery Point Object (RPO)
  - Acceptable data loss measured in time
- Traditional IT model has DR in a second physical site
  - Low end DR: off-size backups
  - High end DR: hot size active-active architecture

amazon web services

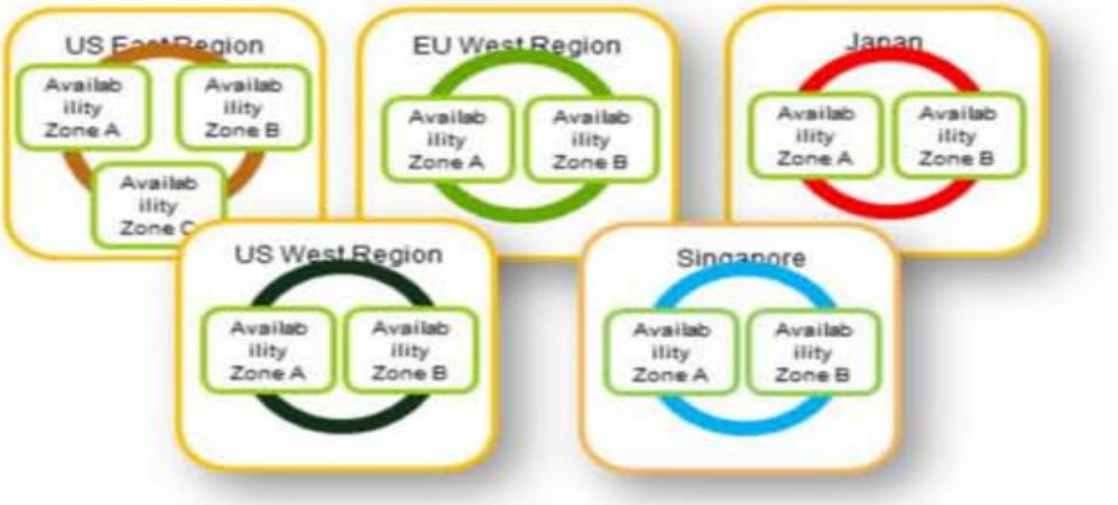
Topics

- How High Availability and Disaster Recovery work together
- Building highly available systems on AWS
- Best practices for high availability and disaster recovery
- Common patterns of disaster recovery on AWS

amazon web services

AWS: Regions and Availability Zones

- Regions are completely separate clouds
- Multi network-connected Availability Zones in each Region



amazon web services

AWS Solutions Architect Associate

High Availability (HA)

High Availability Introduction

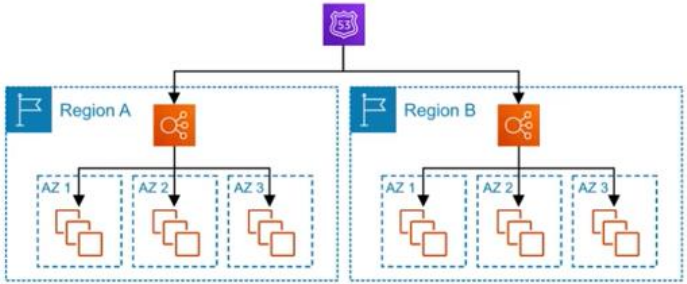
amazon web services

High Availability (HA)

The ability for a system to remain available

Think about what could cause a service to become unavailable:

1. When an AZ becomes unavailable eg. data-center flooded
2. When a Region becomes unavailable eg. meteor strike
3. When an web-application becomes unresponsive eg. too much traffic
4. When an instance becomes unavailable eg. instance failure
5. When a web application becomes unresponsive due to distance in geographic location



The solution we need to implement in order to ensure **High Availability**:

1. We should run our instances in Multi-AZ, an **Elastic Load Balancer** can route traffic to operational AZs.
2. We should run instances in another region. We can route traffic to another Region via **Route53**
3. We should use **Auto Scaling Groups** to increase the amount of instances to meet the demand of traffic
4. We should use **Auto Scaling Groups** to ensure a minimum amount of instances are running and have **ELB** route traffic to healthy instances
5. We should use **CloudFront** to cache static content for faster delivery in nearby regions. We can also run our instances in nearby regions and route traffic using a geolocation policy in **Route53**

amazon web services

AWS Solutions Architect Associate

High Availability (HA)

Scale Up and Scale Out

amazon web services

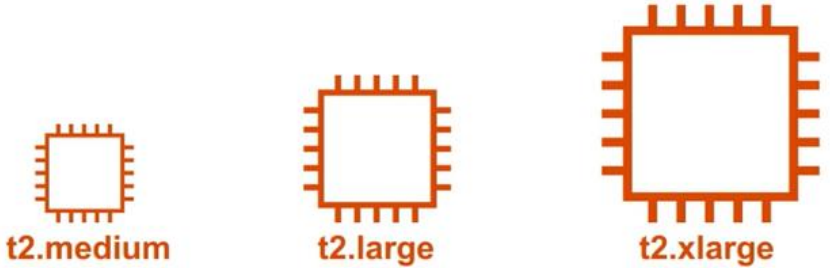
Scale Up vs Scale Out

When utilization increases and we are reaching capacity we can:

Scale up (Vertical Scaling)

Increasing the size of instances

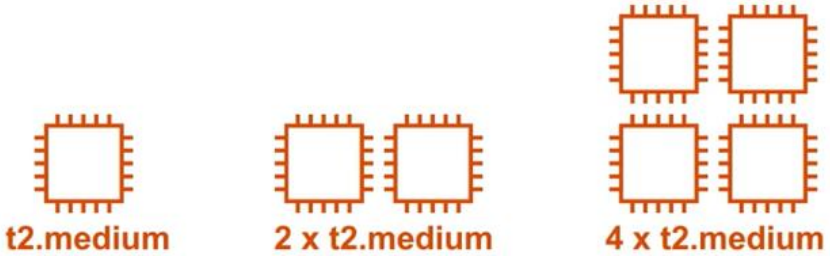
- Simpler to manage.
- Lower availability (if a single instance fails service becomes unavailable)



Scale out (Horizontal Scaling)

Adding more of the same

- More complexity to manage.
- Higher availability (if a single instance fail it doesn't matter)



You will generally want to **scale out** and **then up** to balance complexity vs availability

amazon web services

Take advantage of multiple availability zones

- No cost difference between servers running in a single AZ versus multiple Azs
- Most services are designed with multiple AZ use in mind

amazon web services

Topics

- How High Availability and Disaster Recovery work together
- Building highly available systems on AWS
- Best practices for high availability and disaster recovery
- Common patterns of disaster recovery on AWS

amazon web services

Best Practices for HA

- Build loosely coupled systems
  - –Queues, load-balance tiers
- Implement elasticity
  - –Bootstrapping, load balancing, Auto Scaling, etc…
  - –Instance asks: ”Who am I and what is my role”?
- Use abstract machine and system representations
  - –Build images from recipes, stacks from CloudFormation

amazon web services

Design for failure: Basic principles

- Goal: Applications should continue to function event if the underlying physical hardware fails or is removed or replaced
  - Avoid single points of failure
  - Assume everything fails, and design backwards
  - Design your recovery process

amazon web services

Topic

- How High Availability and Disaster Recovery work together
- Building highly available systems on AWS
- Best practices for high availability and disaster recovery
- Common patterns of disaster recovery on AWS

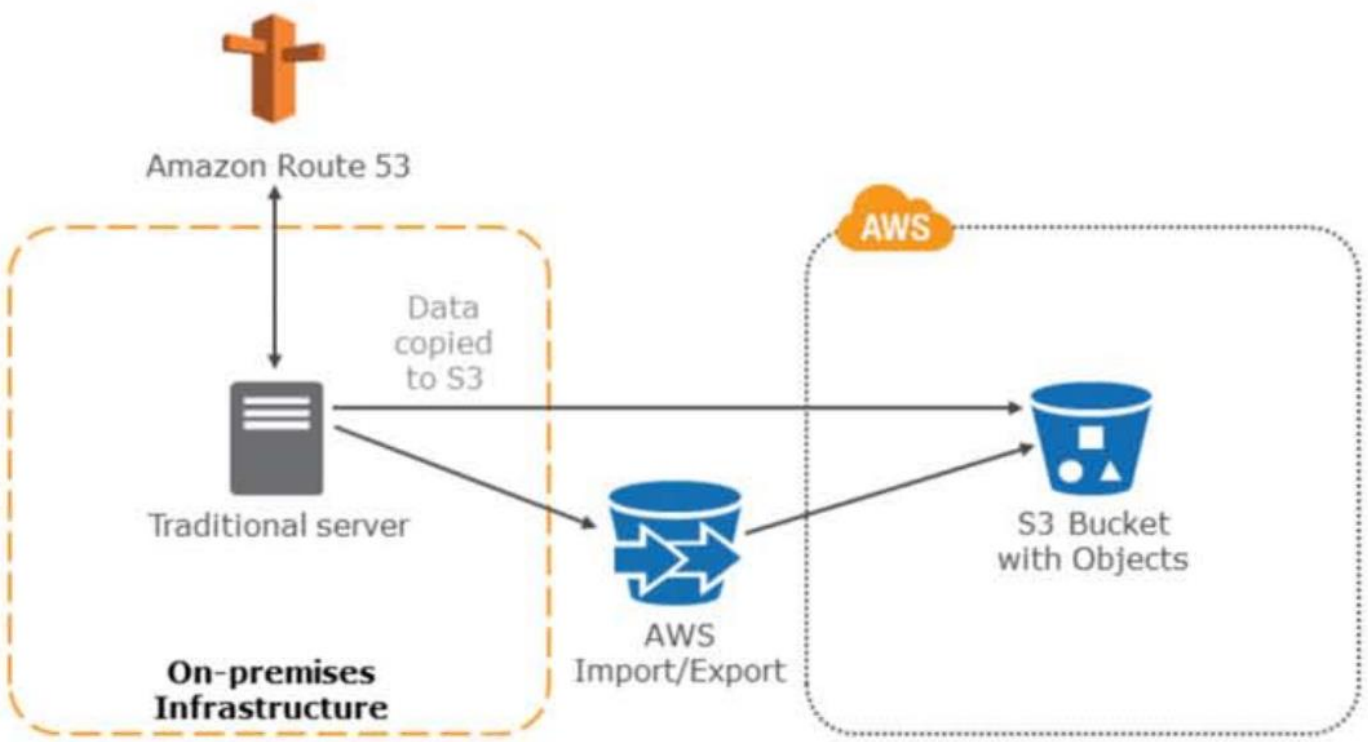
amazon web services

Common Practices of Disaster Recovery on AWS

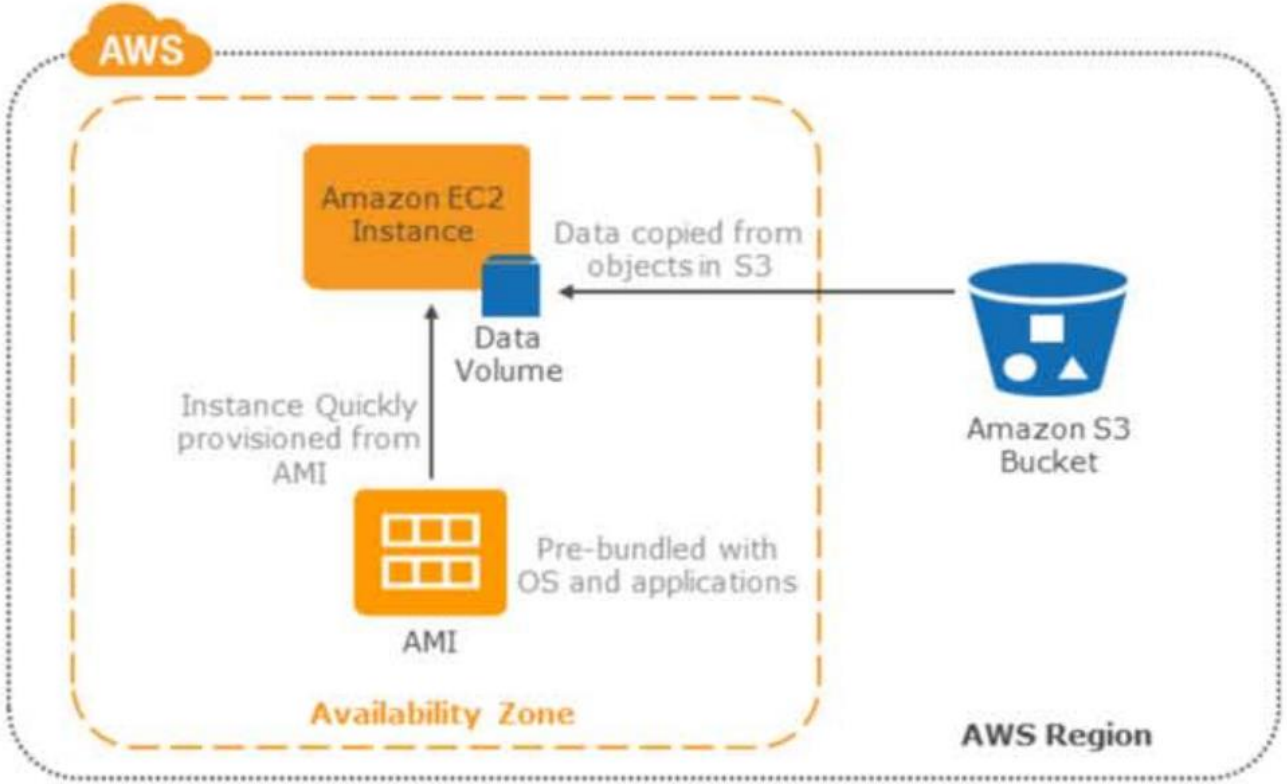
- Example Architectural Patterns (sorted by increasingly optimal RTO/RPO)
  - Backup and Restore
  - Pilot Light
  - Fully Working Low Capacity Standby
  - Multi-Site Hot Standby

amazon web services

Backup and Restore



Backup and Restore



amazon web services

Backup and Restore

- Advantages
  - Simple to get started
  - Extremely cost effective (mostly backup storage)
- Preparation Phase
  - Take backups of current systems
  - Store backups in S3
  - Describe procedure to restore from backups on AWS
    - Know which AMI to use, build your own as needed
    - Know how to restore system from backups
    - Know how to switch to new system
    - Know how to configure the deployment

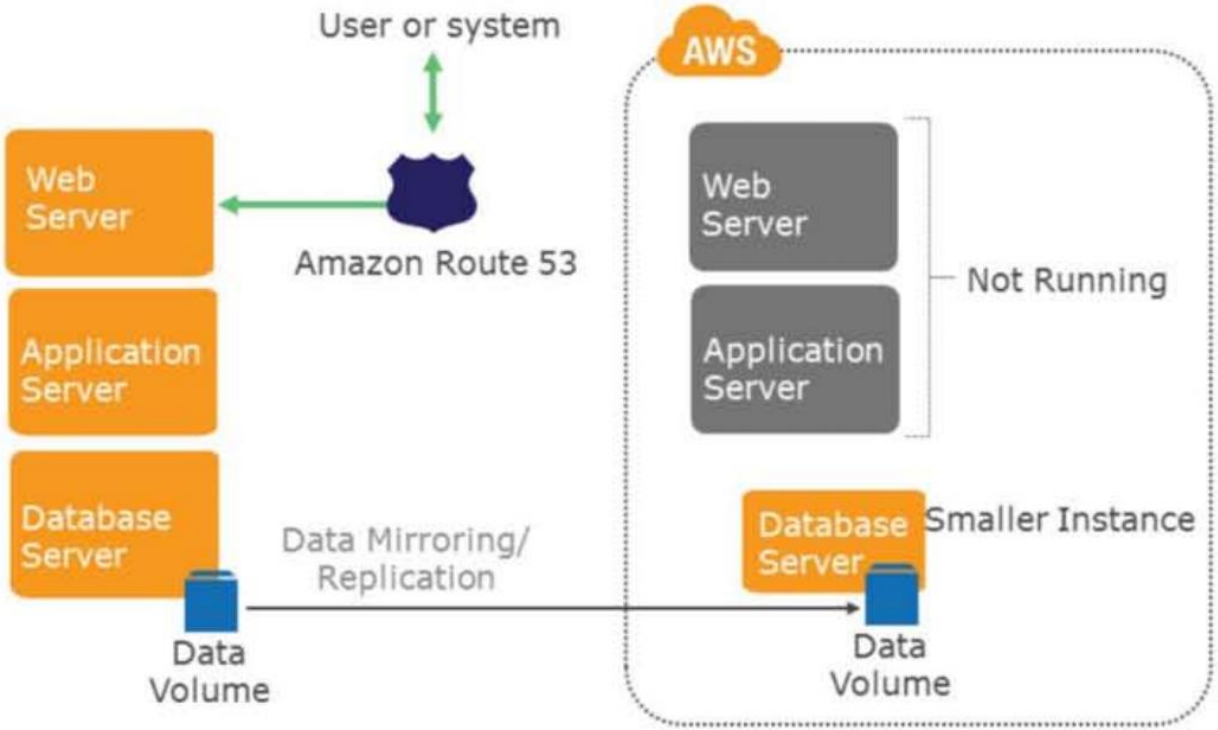
amazon web services

Common Practices of Disaster Recovery on AWS

- In case of disaster
  - Retrieve backups from S3
  - Bring up required infrastructure
    - EC2 instance with prepared AMIs, Load Balancing, etc.
  - Restore system from backup
  - Switch over to the new system
    - Adjust DNS records to point to AWS
- Objectives
  - RTO: as long as it takes to bring up infrastructure and restore system from backups
  - RPO: time since last backup

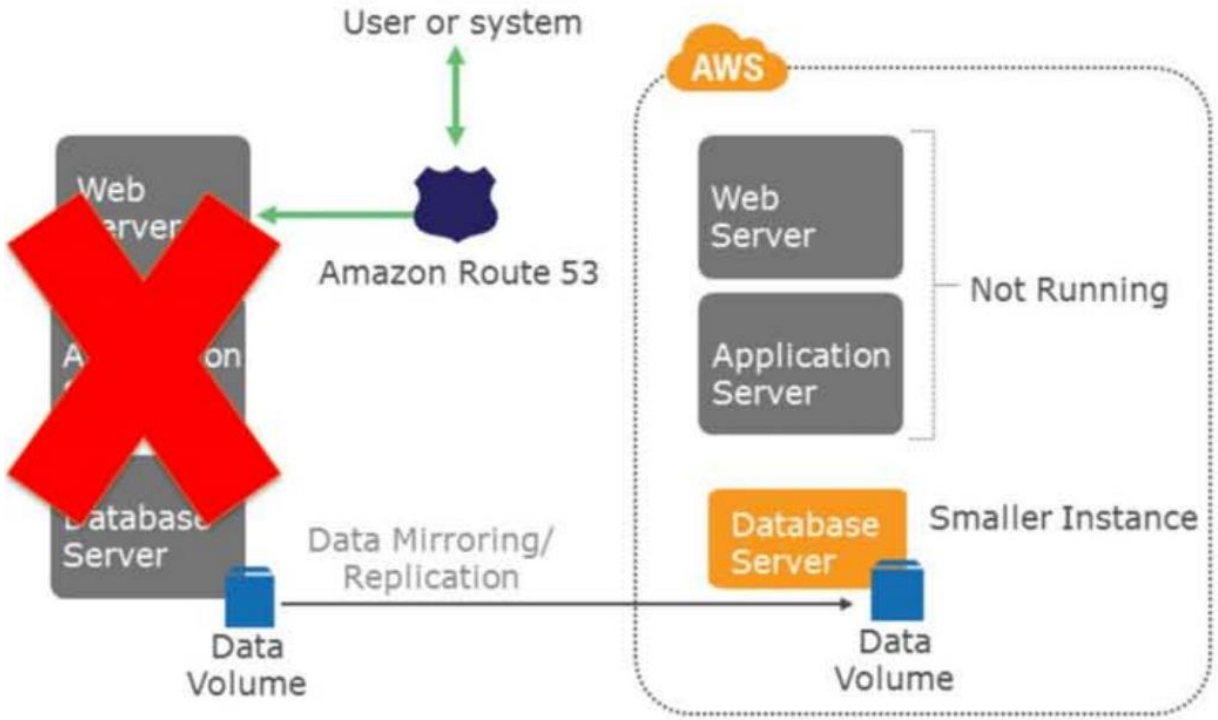
amazon web services

Pilot Light

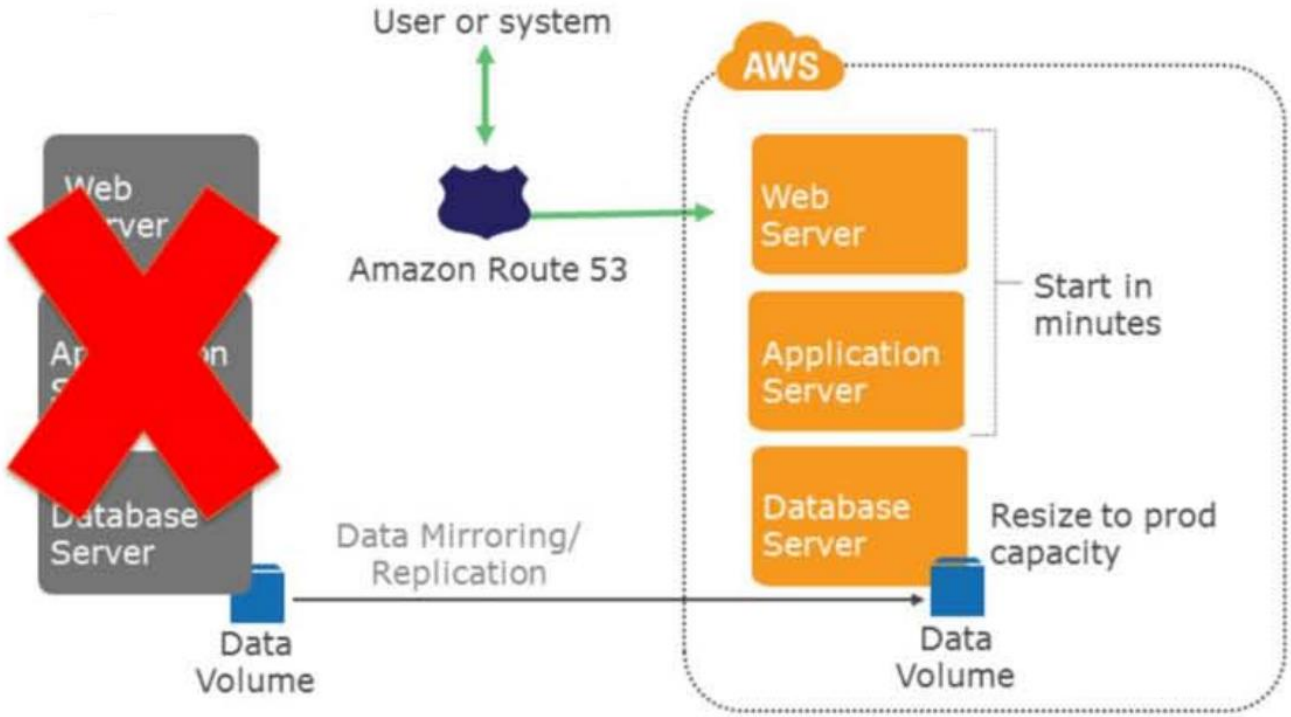


amazon web services

Pilot Light


User or system
AWS
Web Server
Application Server
Not Running
Amazon Route 53
Database Server
Smaller Instance
Data Mirroring/ Replication
Data Volume
Data Volume

Pilot Light


User or system
AWS
Amazon Route 53
Web Server
Application Server
Database Server
Start in minutes
Resize to prod capacity
Data Mirroring/ Replication
Data Volume
Data Volume

Pilot Light

- Advantages
  - Very cost effective (fewer 24/7 resources)
- Preparation Phase
  - Enable replication of all critical data a AWS
  - Prepare all required resources for automatic start
    - AMIs, Network Settings, Load Balancing, ect..
  - Reserved Instances

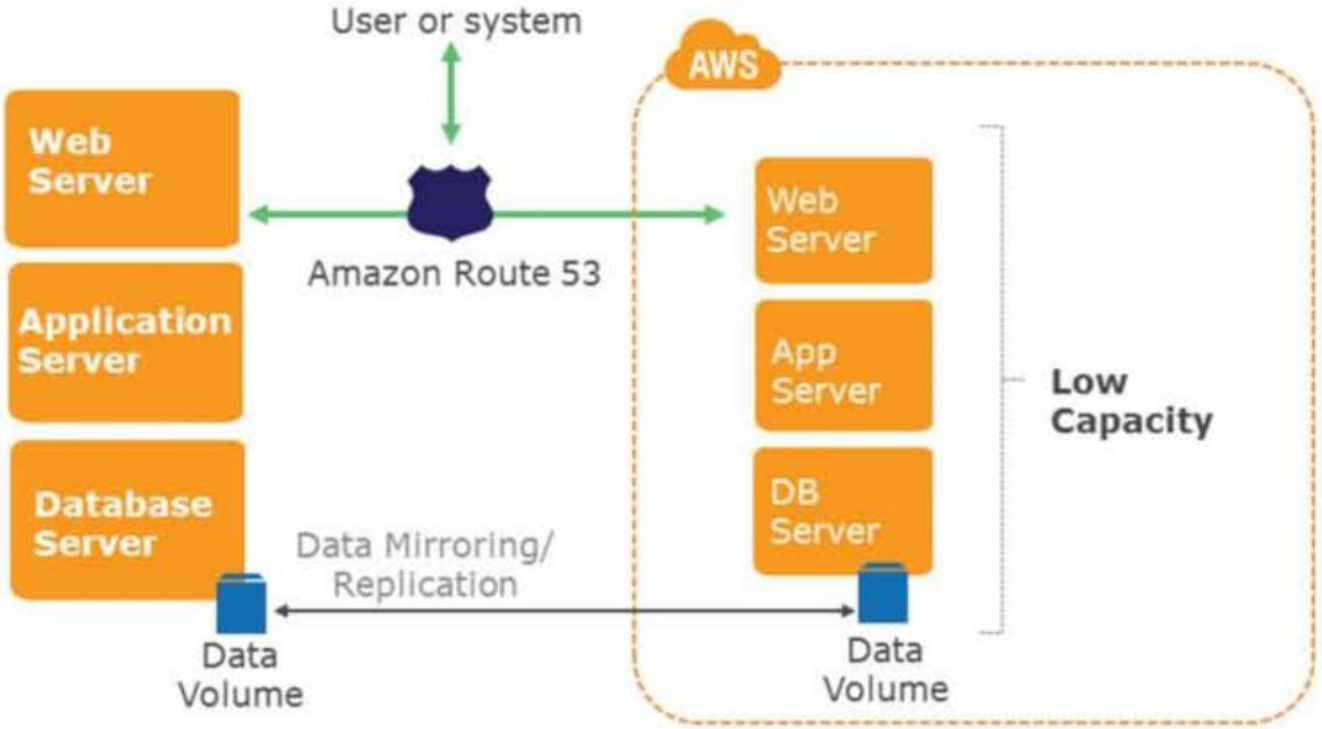
amazon web services

Pilot Light

- In case of disaster
  - –Automatically bring up resources around the replicated core data set
  - –Scale the system as needed to handle current production traffic
  - –Switch over to the new system
    - Adjust DNS records to point to AWS
- Objectives
  - –RTO: as long as it takes to detect need for DR and automatically scale up replacement system
  - –RPO: depends on replication type

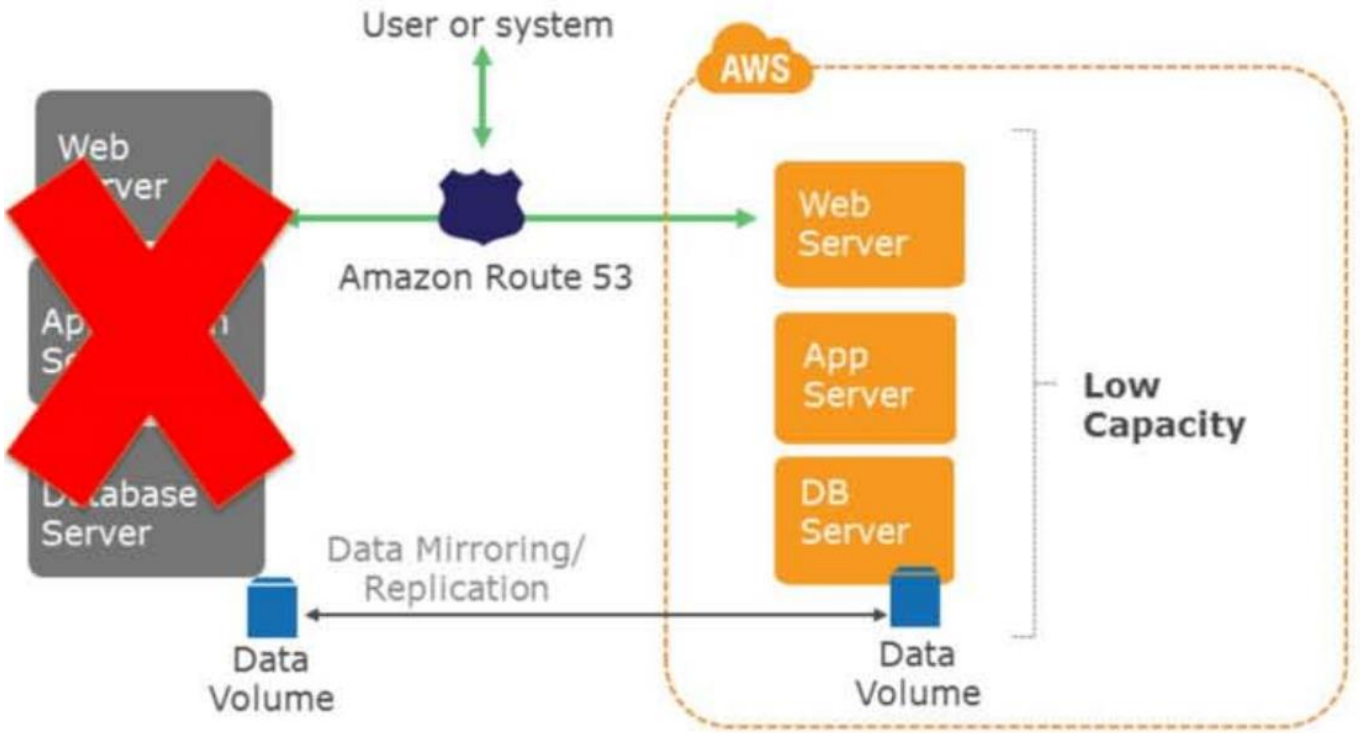
amazon web services

Fully-Working Low Capacity Standby



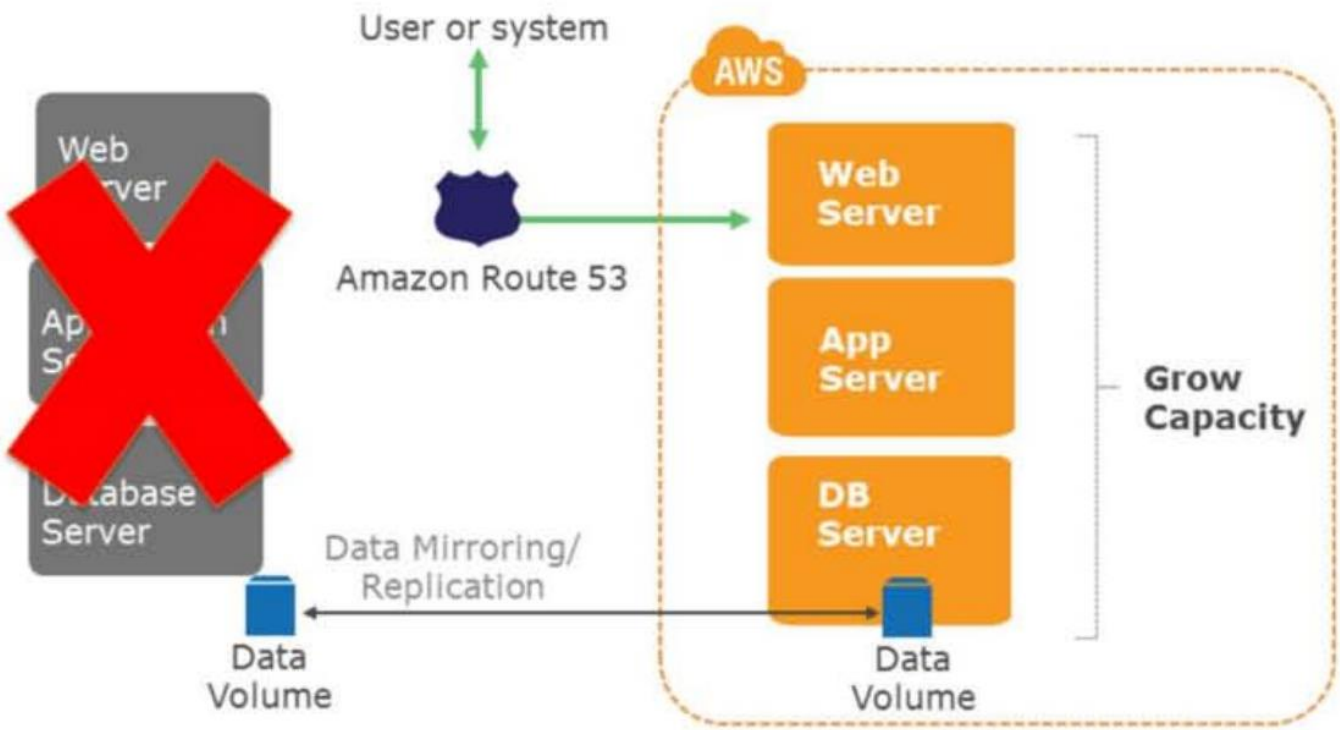
amazon web services

Fully-Working Low Capacity Standby



amazon web services

Fully-Working Low Capacity Standby



amazon web services

Fully-Working Low Capacity Standby

- Advantages
  - –Can take some production traffic at any time
  - –Cost savings (IT footprint smaller than full DR)
- Preparation
  - –Similar to Pilot Light
  - –All necessary components running 24/7, but not scaled for production traffic
  - –Best practice-continuous testing
    - “Trickle” a statistical subset of production traffic to DR site

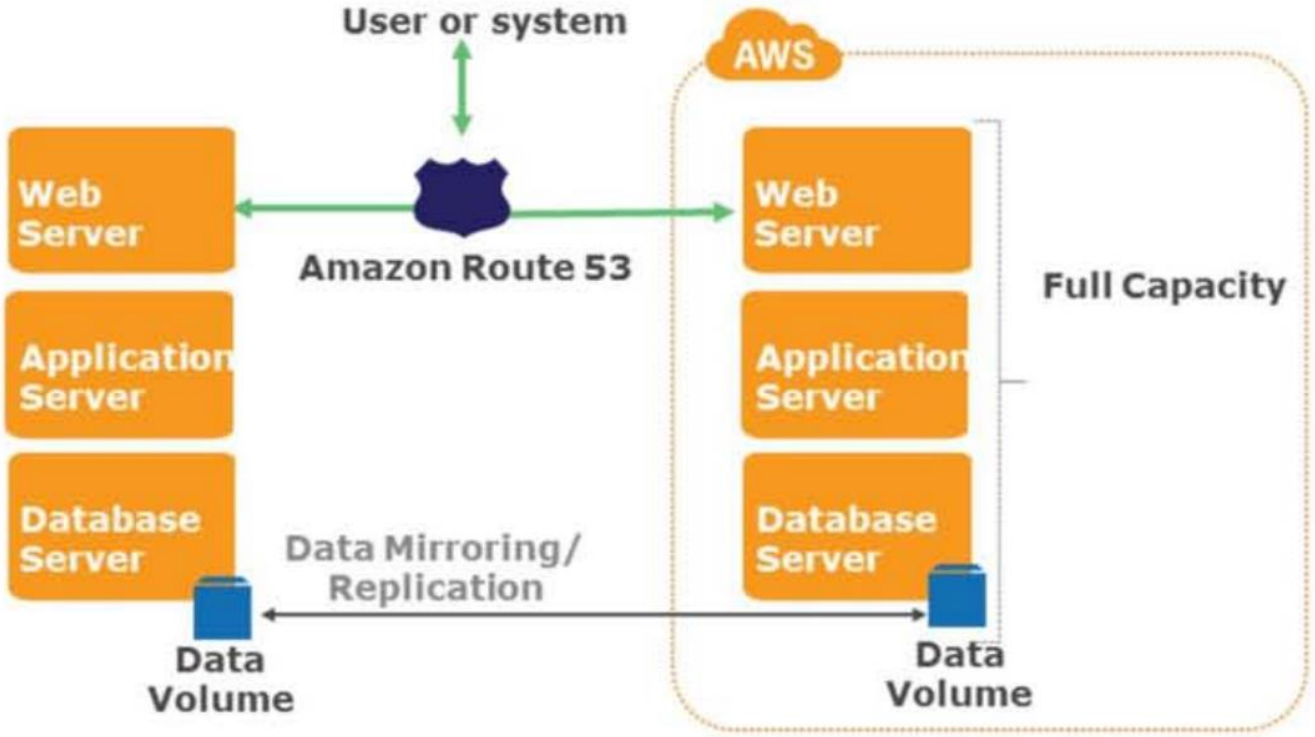
amazon web services

Fully-Working Low Capacity Standby

- In case of disaster
  - Immediately fail over most critical production load
    - Adjust DNS records to point to AWS
  - (Auto) Scale the system further to handle all production load
- Objectives
  - RTO: for critical load: as long as it takes to fail over; for all other load, as long as it takes to scale further
  - RPO: depends on replication type

amazon web services

Multi-site Active-Active



amazon web services

Multi-site Active-Active

- Advantages
  - At any moment can take all production load
- Preparation
  - Similar to Low-Capacity Standby
  - Fully scaling in/out with production load
- In case of Disaster
  - Immediately fail over all production load
    - Adjust DNS records to points to AWS
- Objectives
  - RTO: as long as it takes fail over
  - RPO: depends on replication type

amazon web services

Hosted Desktop

- Advantages
  - –Replacement of workstations in case of disaster
  - –Pay only when used for DR
- Preparation
  - –Set up AMIs with appropriate working environment
- In Case of Disaster
  - –Launch desktop AMI and resume work
- Objectives
  - –RTO: as long as it takes to launch AMI and restore work environment on virtual desktop
  - –RPO: depends on state of AMI

amazon web services

Best Practices for Being Prepared

- Start simple and work your way up
  - Backups in AWS as first step
  - Incrementally improve RTO/RPO as continuous effort
- Check for any software licensing issues
- Exercise your DR Solution
  - Game Day
  - Ensure backups, snapshots, AMIs, etc.. Are working
  - Monitor your monitoring system

amazon web services

Conclusion – Advantages of disaster recovery with AWS

- Various building blocks available
- Fine control overs cost vs. RTO/RPO tradeoffs
- Ability to scale up when needed
- Pay for what you use, and only when you use it(when an event happens)
- Ability to easily and effectively test your DR plan
- Availability of multiple locations world wide
- Hosted desktops available
- Variety of Solution Providers

amazon web services

Putting It Together: Incremental Improvement

- Start with existing on-premise app with traditional DR
- Gradually move DR (and thus the app) to the cloud:
  - Backup to S3 and Restore
  - "Pilot Light" on AWS for Quick Recovery
  - Fully Working Low Capacity Standby on AWS
- Migrate to primary on AWS, DR on-premise
- Add hot standby in second AWS AZ or Region
- Incrementally add HA features to primary app

amazon web services