aws academy

CSCI 5902 Adv. Cloud Architecting
Fall 2023
Instructor: Dr. Lu Yang

Module 8 Securing User and Application Access (Sections 2-5) & Module 9 Implementing Elasticity, High Availability, and Monitoring (Sections 1-2)
Nov 6, 2023

© 2020 Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Housekeeping items and feedback

1. Start recording
2. Start working on your term project.
3. I will make a video this weekend to explain the questions in the midterm.

AWS Academy Cloud Architecting

Module 8: Securing User and Application Access

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module overview

Sections

1. Architectural need
2. Account users and IAM
3. Organizing users
4. Federating users
5. Multiple accounts

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Externally authenticated users

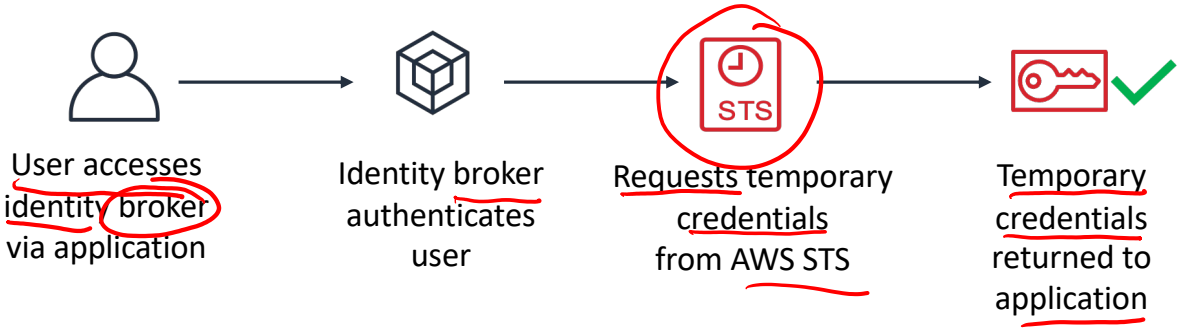
Identity federation

- User authentication completed by a system that is external to the AWS account
  - Example: corporate directory
- It provides a way to allow access through existing identities, without creating IAM users

Identity federation options

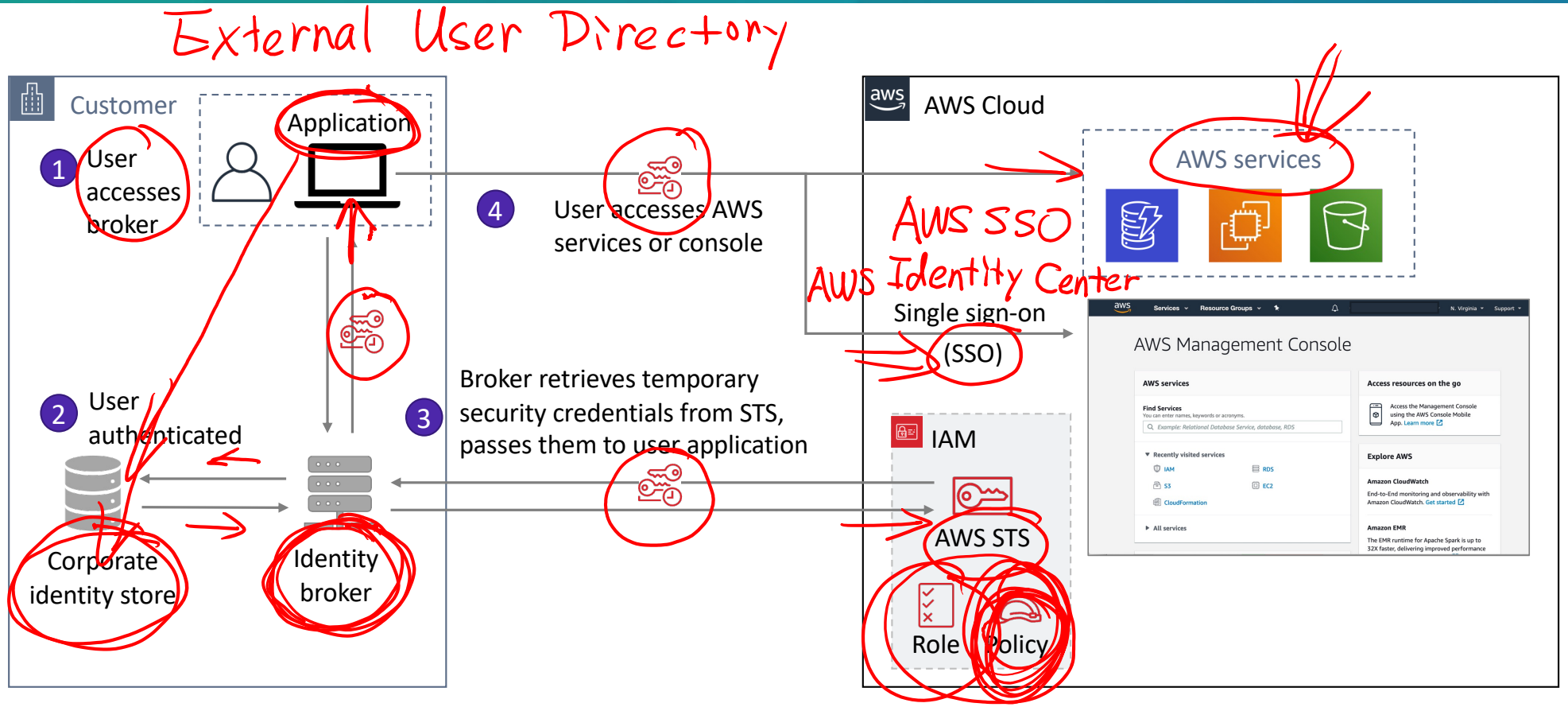
1. AWS STS
   - Public identity service providers (IdPs)
   - Custom identity broker application
2. Security Assertion Markup Language (SAML)
3. Amazon Cognito

IdP authentication overview



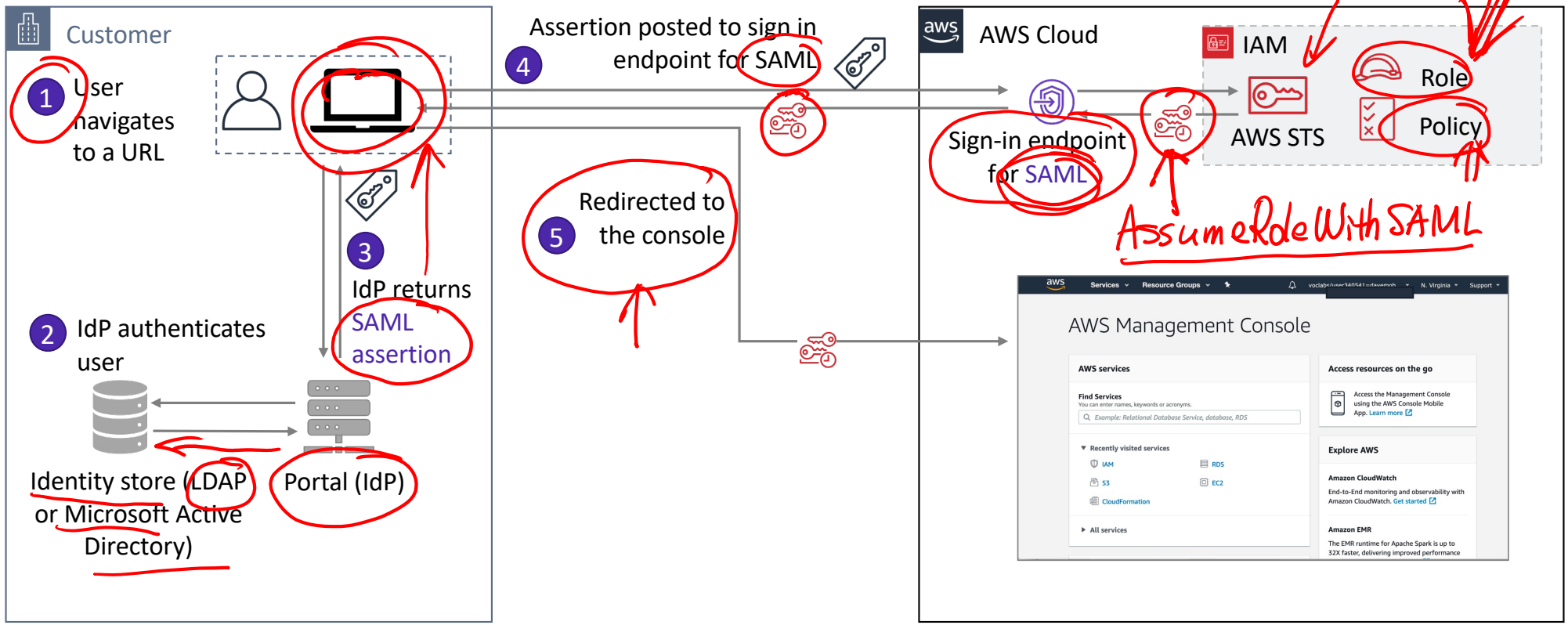
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

1. Identity federation with an identity broker



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Identity federation using SAML



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

3. Amazon Cognito

Amazon Cognito is a fully managed service.

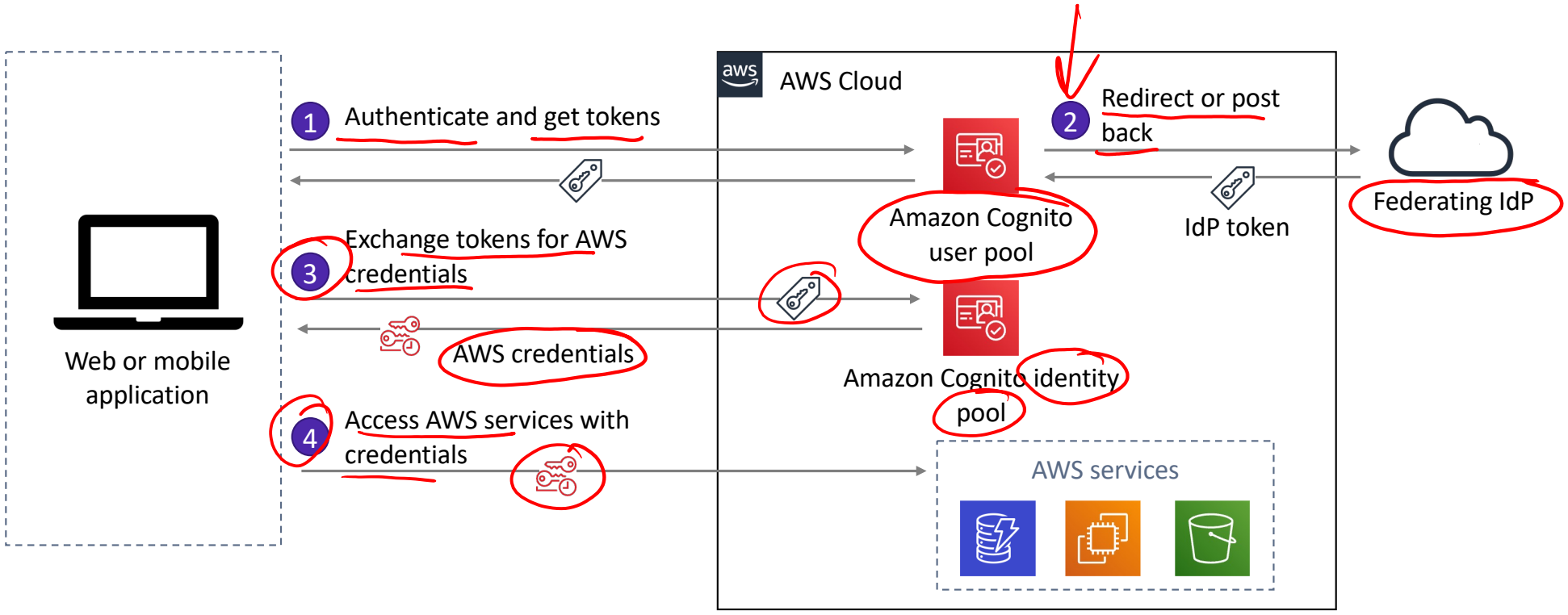
- It provides authentication, authorization, and user management for web and mobile applications
- Amazon Cognito provides web identity federation
  - They can be used as the identity broker that supports IdPs that are compatible with OpenID Connect (OIDC)
- Federated identities
  - Users sign in with social identity providers (Amazon, Facebook, Google) or with SAML
- User pools
  - You can maintain a directory with user profiles authentication tokens

① new users
② federate
③ who

• Identity Pools ← permissions

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon Cognito example



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Section 4 key takeaways

- IAM roles provide temporary security credentials assumable by a person, application, or service
- The AWS Security Token Service (AWS STS) enables you to request temporary AWS credentials
- With identity federation, user authentication is external to the AWS account
  - Accomplished by using AWS STS, SAML, or Amazon Cognito

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module 8: Securing User and Application Access

Section 5: Multiple accounts

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws academy

One account or multiple accounts?

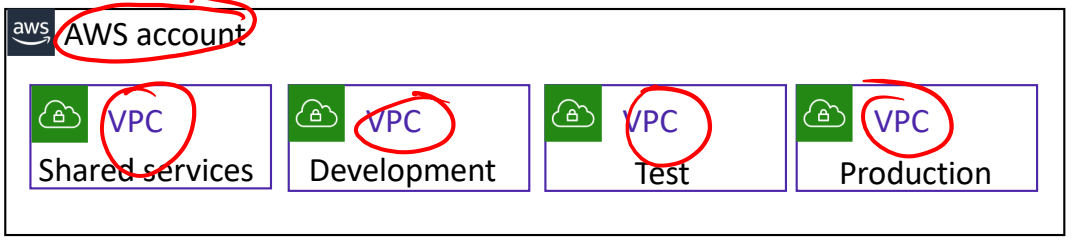
Two architectural patterns

- Most organizations choose to create multiple accounts

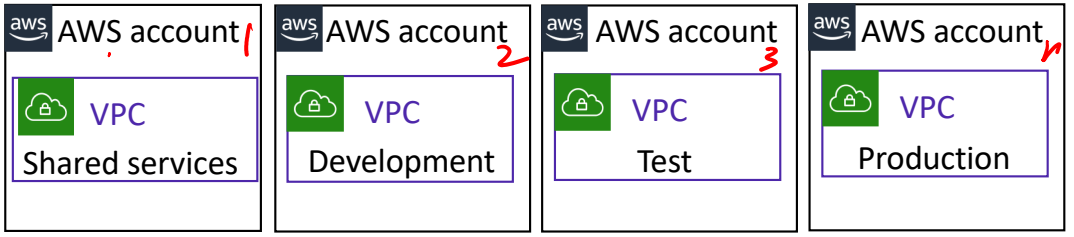
Advantages of multiple accounts

- Isolate business units or departments
- Isolate development, test, and production environments
- Isolate auditing data, recovery data
- Separate accounts for regulated workloads
- Easier to trigger cost alerts for each business unit's consumption

Multiple VPCs in a single account architectural pattern



Multiple accounts, a VPC in each account architectural pattern



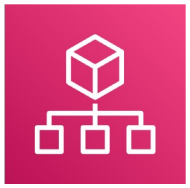
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Challenges for managing multiple accounts

- Security management across accounts
  - IAM policy replication
- Creating new accounts
  - Involves many manual processes
- Billing consolidation
- Centralized governance is needed to ensure consistency

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Manage multiple accounts with AWS Organizations



AWS Organizations

Centrally manage and enforce policies across multiple AWS accounts.

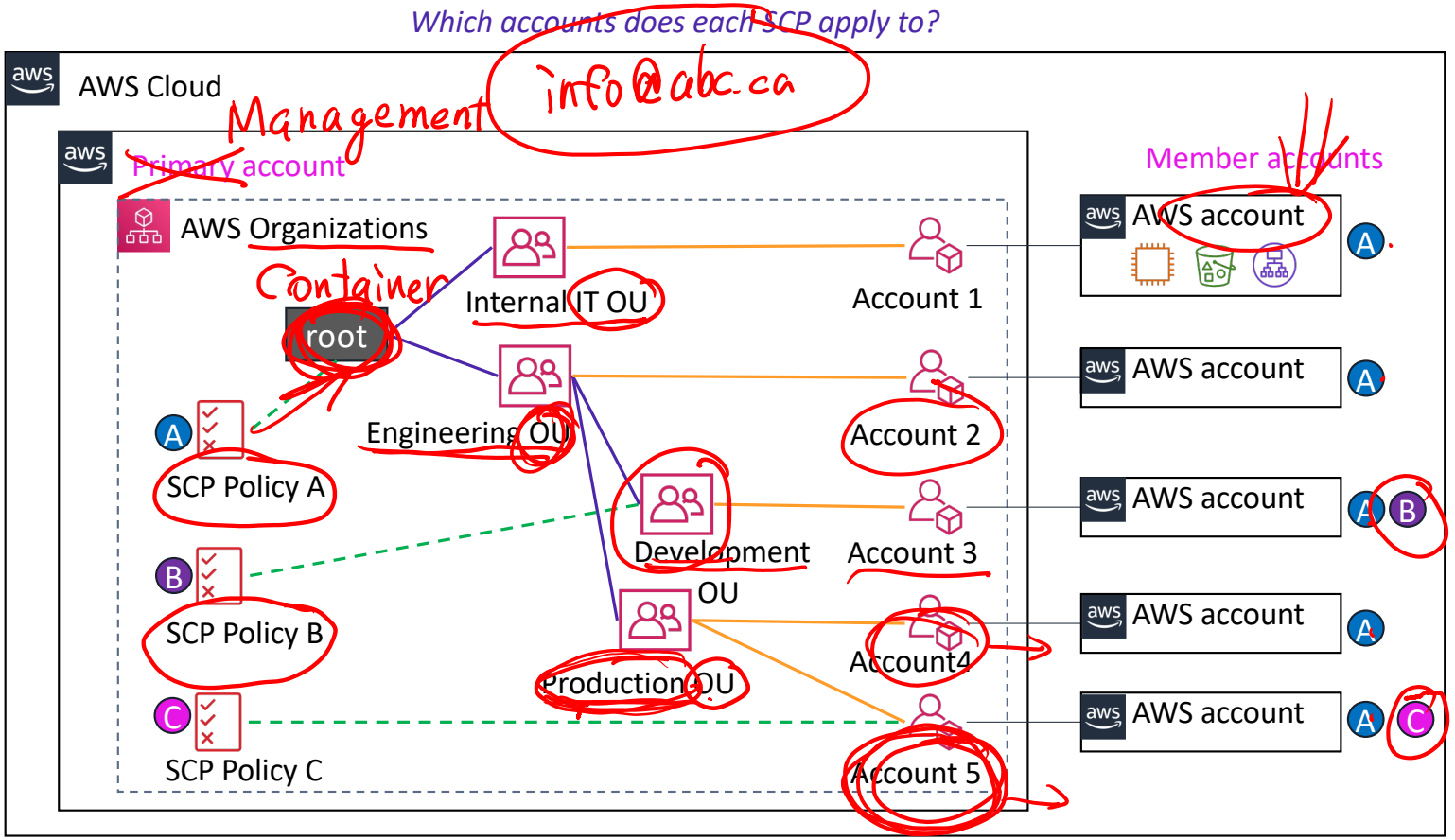
- Group-based account management
- Policy-based access to AWS services
- Automated account creation and management
- Consolidated billing
- API-based

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWS Organizations: Illustrated

In the AWS Organizations primary account:

1. Create a hierarchy of organizational units (OUs)
2. Assign accounts to OUs as member accounts
3. Define service control policies (SCPs) that apply permissions restrictions to specific member accounts
4. Attach the SPCs to root, OUs, or accounts



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Example uses of SCPs

- Characteristics of service control policies (SCPs)
  - They enable you to control which services are accessible to IAM users in member accounts
  - IAM policies that are defined in individual accounts still apply
- Example uses of SCPs
  - Create a policy that *blocks* service access or specific actions

    Example: Deny users from disabling AWS CloudTrail in all member accounts
  - Create a policy that *allows* full access to specific services

    Example: Allow full access to Amazon EC2 and CloudWatch
  - Create a policy that *enforces the tagging* of resources

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Section 5 key takeaways

- You can use multiple AWS accounts to isolate business units, development and test environments, regulated workloads, and auditing data
- AWS Organizations enables you to configure automated account creation and consolidated billing
- You can configure access controls across accounts by using service control policies (SCPs)

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module 8: Securing User and Application Access

Module wrap-up

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws academy

Module summary

In summary, in this module, you learned how to:

- Explain the purpose of AWS Identity and Access Management (IAM) users, groups, and roles
- Describe how to allow user federation within an architecture to increase security
- Recognize how AWS Organizations service control policies (SCPs) increase security within an architecture
- Describe how to manage multiple AWS accounts
- Configure IAM users

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

AWS Academy Cloud Architecting

Module 9: Implementing Elasticity, High Availability, and Monitoring

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module overview

aws academy

Sections

1. Architectural need
2. Scaling your compute resources
3. Scaling your databases
4. Designing an environment that's highly available
5. Monitoring

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module objectives

At the end of this module, you should be able to:

- Use Amazon EC2 Auto Scaling within an architecture to promote elasticity
- Explain how to scale your database resources
- Deploy an Application Load Balancer to create a highly available environment
- Use Amazon Route 53 for Domain Name System (DNS) failover
- Create a highly available environment
- Design architectures that use Amazon CloudWatch to monitor resources and react accordingly

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

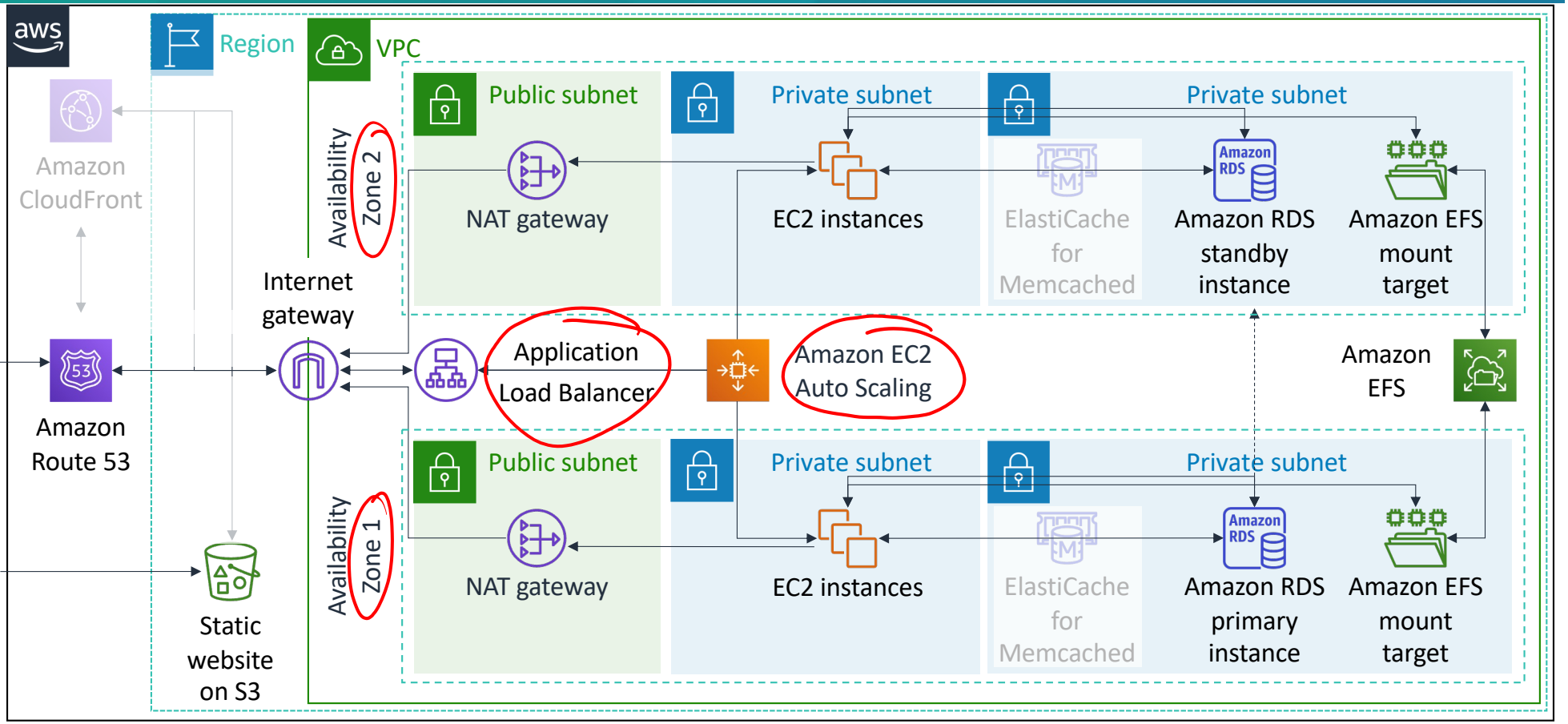
Module 9: Implementing Elasticity, High Availability, and Monitoring

Section 1: Architectural need

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws academy

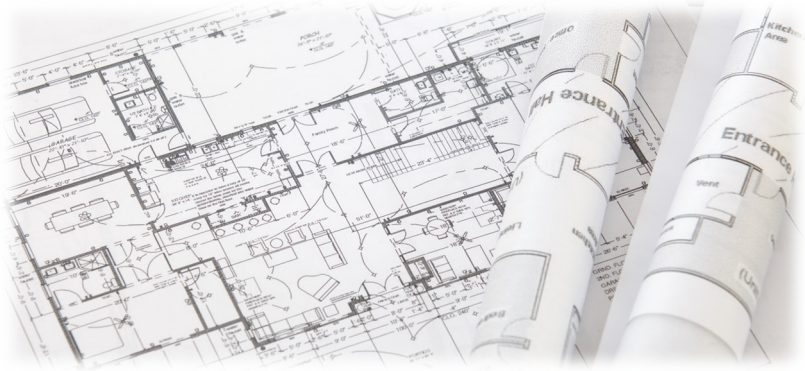
Implementing high availability as part of a larger architecture



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Café business requirement

The café will be featured in a famous TV food show. When it airs, the architecture must handle significant increases in capacity.



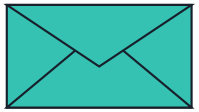
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Reactive architectures

Elastic
and scalable

Resilient

Responsive



Message-driven

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Module 9: Implementing Elasticity, High Availability, and Monitoring

Section 2: Scaling your compute resources

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

aws academy

What is elasticity?

An elastic infrastructure can expand and contract as capacity needs change.

Examples:

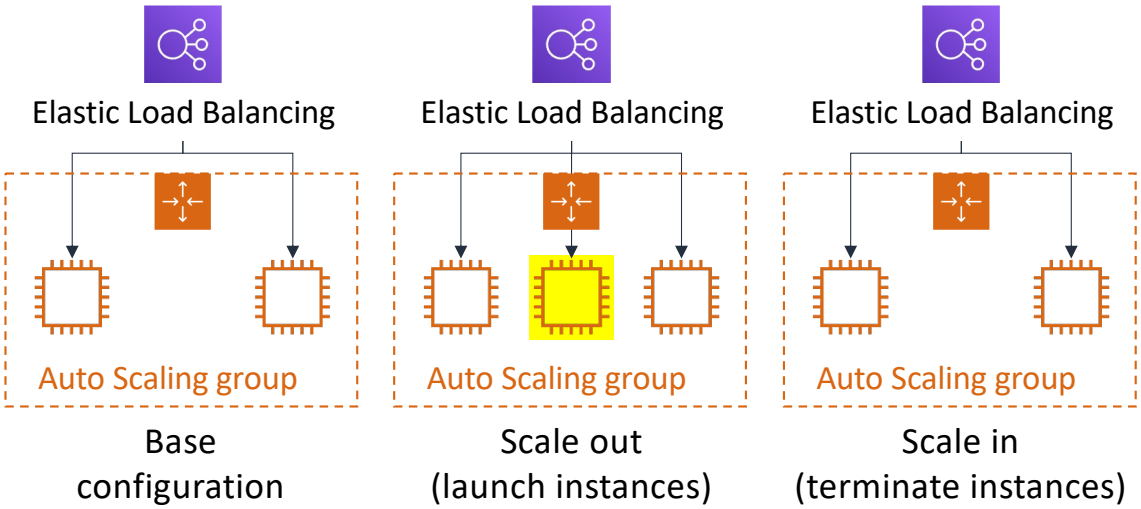
- Increasing the number of web servers when traffic spikes
- Lowering write capacity on your database when traffic goes down
- Handling the day-to-day fluctuation of demand throughout your architecture

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

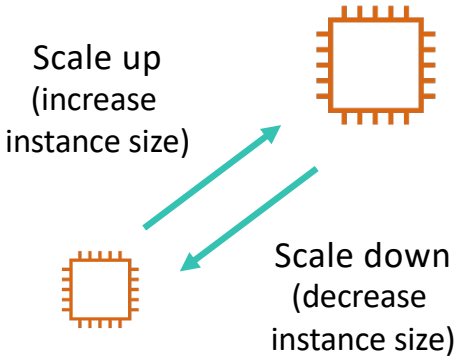
What is scaling?

A technique that is used to achieve elasticity

Horizontal scaling



Vertical scaling



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Amazon EC2 Auto Scaling

Amazon EC2
Auto Scaling

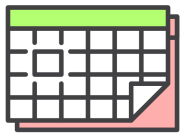
- Launches or terminates instances based on specified conditions
- Automatically registers new instances with load balancers when specified
- Can launch across Availability Zones

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Scaling options

1 Scheduled

Good for predictable workloads



Scale based on date and time

Use case: Turning off your development and test instances at night

2 Dynamic

Good for changing conditions

Supports target tracking

Use case: Scaling based on CPU utilization

3 Predictive

Good for predicted demand

Scale based on machine learning

Use case: Handling an increase in workload for ecommerce website during a major sales event

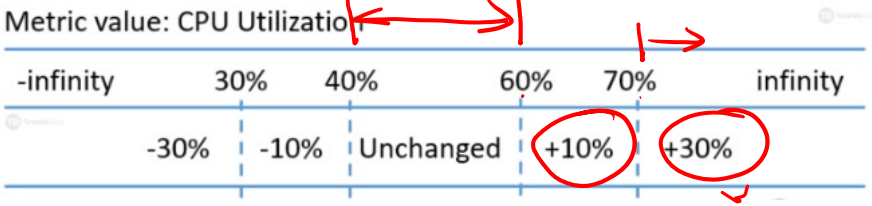
© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Dynamic scaling policy types

- Simple scaling – Single scaling adjustment
  - Use cases: New workloads, spiky workloads

  For example, you can set a CloudWatch alarm to have a CPU Utilization threshold of 80%, and then set the scaling policy to add 20% more capacity to your Auto Scaling group by launching new instances. Accordingly, you can also set a CloudWatch alarm to have a CPU utilization threshold of 30%. When the threshold is met, the Auto Scaling group will remove 20% of its capacity by terminating EC2 instances.

- Step scaling – Adjustment depends on size of alarm breach
  - Use case: Further refines the simple scaling



- Target tracking scaling – Target value for specific metric
  - Use case: Horizontally scalable applications, such as load-balanced and batch data-processing applications

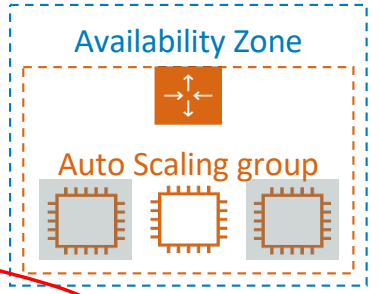
  For example, your scaling metric is the average CPU utilization, and that their average should always be 80%. When CloudWatch detects that it is beyond 80%, it will trigger your target tracking policy to scale out the auto scaling group to meet this target utilization. Once it has gone below 80%, another scale in action will kick in and reduce the number of auto scaling instances in your auto scaling group.

Auto Scaling Groups

An Auto Scaling Group defines:

- Minimum capacity
- Maximum capacity
- Desired capacity*

Capacity?



*The desired capacity reflects the number of instances that are running and can fluctuate in response to events.

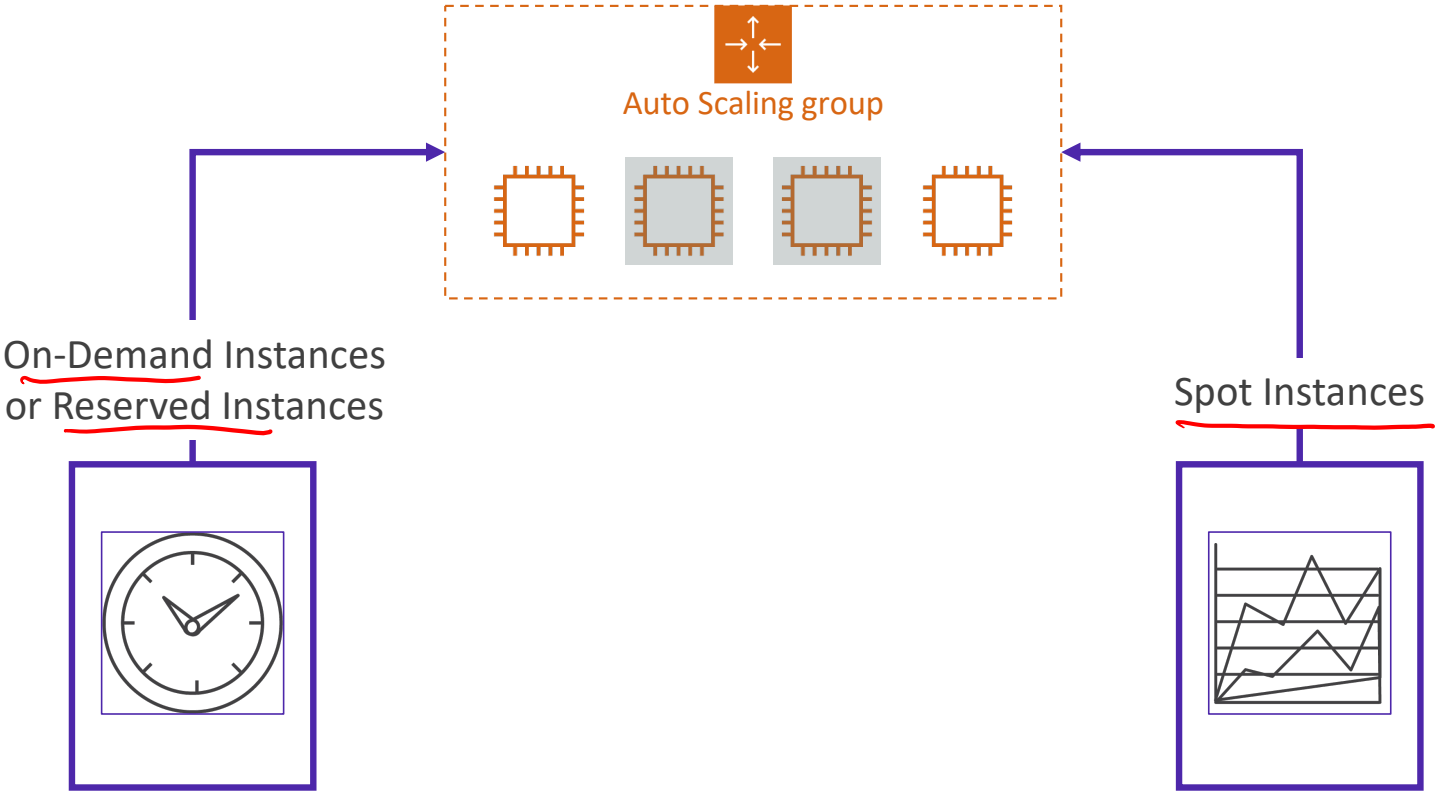
"Desired" is (necessarily) ambiguous

•It means the "**initial**" number of instances. Why not just "initial" then? Because the number may change by autoscaling events.

•So it means "**current**" number of instance. Why not just "current" then? Because during an autoscaling event, instances will start / terminate. Those instances do not count towards "current" number of instances. By "current", a user expects instances that are operate-able.

•So it means "**target**" number of instance. Why not just "target" then? I guess "target" is just as good (ambiguous) as "desired"...

Amazon EC2 Auto Scaling: Purchasing options



© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Automatic scaling considerations

- Multiple types of automatic scaling
- Simple, step, or target tracking scaling — Dynamic
- Multiple metrics (not just CPU)
  - Metrics that decrease when capacity increases—and increase when capacity decreases—can be used to proportionally scale out or in the number of instances that use target tracking.
- When to scale out and scale in
- Use of lifecycle hooks
  - Try to scale out early and fast, and scale in slowly over time

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Demonstration: Creating Scaling Policies for Amazon EC2 Auto Scaling

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Cloud Solutions Architecture Discussions

Highly Available EC2 Architecture: A Stateless Web Application

A stateless application is one that needs **no knowledge** of **previous interactions** and stores **no session information**.

https://www.youtube.com/watch?v=LRfpgJMjcSg

Section 2 key takeaways

- An elastic infrastructure can expand and contract as capacity needs change
- Amazon EC2 Auto Scaling automatically adds or removes EC2 instances according to policies that you define, schedules, and health checks
- Amazon EC2 Auto Scaling provides several scaling options to best meet the needs of your applications
- When you configure an Auto Scaling group, you can specify the EC2 instance types and the combination of pricing models that it uses

© 2020, Amazon Web Services, Inc. or its Affiliates. All rights reserved.

Thank you, and Kahoot!

© 2020 Amazon Web Services, Inc. or its affiliates. All rights reserved. This work may not be reproduced or redistributed, in whole or in part, without prior written permission from Amazon Web Services, Inc. Commercial copying, lending, or selling is prohibited. Corrections or feedback on the course, please email us at: aws-course-feedback@amazon.com. For all other questions, contact us at: https://aws.amazon.com/contact-us/aws-training/. All trademarks are the property of their owners.

aws academy