3. Cloud application

Updates

- **3 Oct 2025:** This assessment is worth 40% of your final grade. It will be marked out of 45 points, with scores scaled accordingly.

Table of Contents

Overview

Task description

You will complete your cloud project by migrating the processing tasks of your application to a cloud architecture, that makes use of multiple cloud compute services, designed to be cost effective at a global scale. You will be required to write a professional report describing your work.

This assessment item is eligible for the 48 hour extension. You can read more information about this on the HiQ website (https://qutvirtual4.qut.edu.au/group/student/study/assignments/submitting/late-assignments-and-extensions) and the recent CAB432 announcement, "Clarifications on the 48-hour late submission period and formal extensions" (https://canvas.qut.edu.au/courses/20367/discussion_topics/405158) .

Unit Learning Outcomes assessed

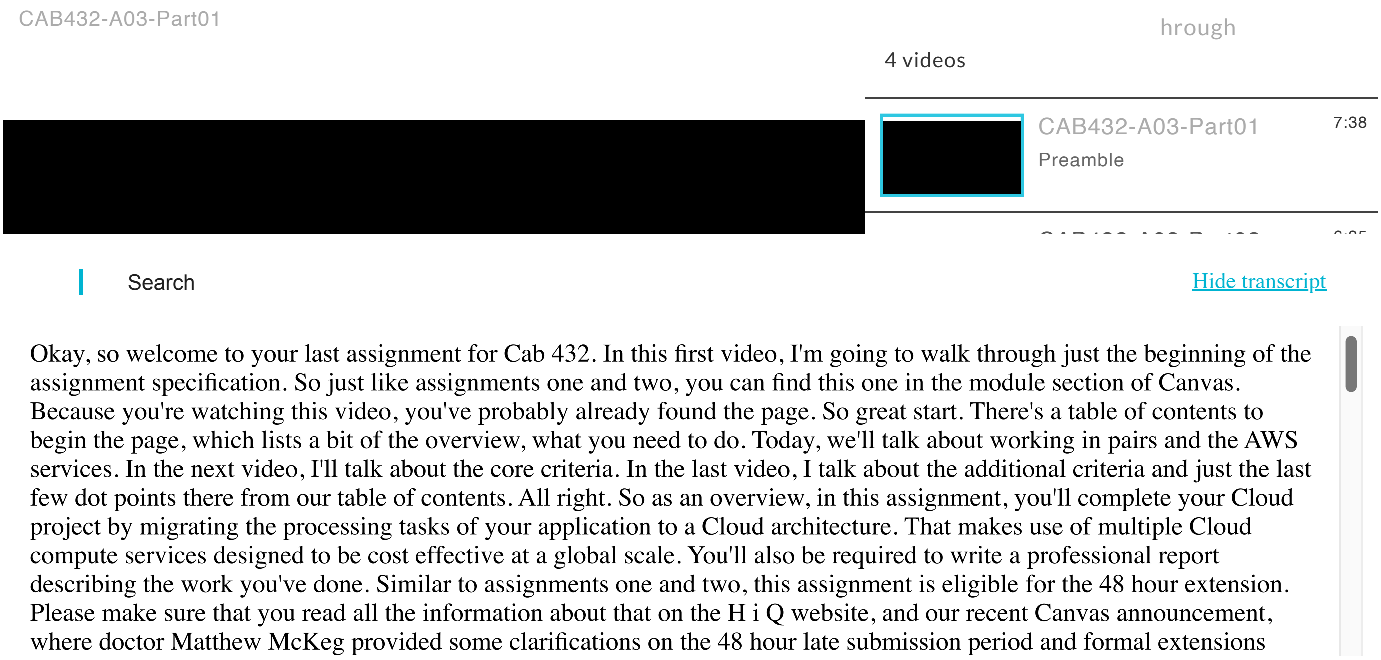
- ULO1 Discuss the elastic nature of cloud technologies and business models and their application in both technical and commercial contexts.
- ULO2 Analyse application design and implementation requirements to select suitable cloud infrastructure and software services from a range of XaaS offerings.
- ULO3 Critically analyse the effectiveness of cloud architectures at scale – encompassing computation, persistence, scaling, statelessness, security privacy and cost.
- ULO4 Design and implement scalable cloud applications using industry standard languages and APIs, deployed on a public cloud infrastructure and leveraging a range of cloud services.
- ULO5 Communicate in written forms about cloud applications, services, and architectures.

Weighting	Group or Individual	How I will be assessed
40% of final grade	Individual or pairs	Grade out of 40

Talk-through videos

In these videos, Jake talks through the assignment.

Note: these videos were recorded in the days leading up to the release of the assignment. If there have since been changes, the videos will differ and won't be updated. Consider the contents of this page to be the most accurate source of information.



Okay, so welcome to your last assignment for Cab 432. In this first video, I'm going to walk through just the beginning of the assignment specification. So just like assignments one and two, you can find this one in the module section of Canvas. Because you're watching this video, you've probably already found the page. So great start. There's a table of contents to begin the page, which lists a bit of the overview, what you need to do. Today, we'll talk about working in pairs and the AWS services. In the next video, I'll talk about the core criteria. In the last video, I talk about the additional criteria and just the last few dot points there from our table of contents. All right. So as an overview, in this assignment, you'll complete your Cloud project by migrating the processing tasks of your application to a Cloud architecture. That makes use of multiple Cloud compute services designed to be cost effective at a global scale. You'll also be required to write a professional report describing the work you've done. Similar to assignments one and two, this assignment is eligible for the 48 hour extension. Please make sure that you read all the information about that on the H i Q website, and our recent Canvas announcement, where doctor Matthew McKeg provided some clarifications on the 48 hour late submission period and formal extensions

What you need to do

This assessment item's practical component concentrates on horizontal scalability, cloud architecture, and communication patterns. You'll submit your code along with a short video that demonstrates your project's functionality, similar to previous assessment items. Additionally, you will write a report that discusses motivations for your project implementation choices and addresses additional aspects of your project including security and sustainability.

You are free to add additional functionality to your application or to replace your application. Aside from where those might interact with the criteria that we specify below such changes won't affect your grade for this assessment item. However, if your project didn't meet the core requirements in Assessment 2 then some such changes may be required to accommodate the development required for this assessment.

Working in pairs

For this assessment item you have the *option* of working in pairs. The complexity of your project will continue to increase; working with a partner will help to compensate for that additional complexity.

- You are free to work on your own if that is your preference. We will apply the marking criteria the same for individuals and pairs.
- If you worked with a partner for Assessment 2 then you can continue to work with them, change partners, or work on your own for this assessment.
- You can choose to work from either partner's Assessment 2 as the starting point, you can combine both partners' projects in some way, or you can start from scratch.
- If one partner obtains a formal extension for this assessment item then it will automatically apply to the other partner.
- We have created the *Find partners for assessments 2 and 3* channel on Teams to help you find a partner. We suggest that you ensure that you know:
  - what grade your partner is aiming for
  - how and when your partner can meet to work on the assessment
  - what skills your partner has that can complement your own

Please sign up your pair on Canvas (but not if working alone). **You need sign up separately for Assessment 3, even if you already signed up to a group for Assessment 2.**

- Go to the *People* tab on Canvas on the sidebar
- Find a group like *A3-Pairs N* that has no students in it
- Sign up for the group
- Get your partner to sign up to the same group

If you accidentally sign up for the wrong group then email cab432@qut.edu.au and ask to be removed from the group. **Please provide the group name/number**.

Please don't sign up for a group if you are working alone.

AWS services and working in pairs

In the CAB432 AWS account some services are tied to you username and special steps need to be taken to give your partner access. See **AWS working with a partner on AWS services (https://canvas.qut.edu.au/courses/20367/pages/working-with-a-partner-on-aws-services)** for more information.

In this assignment, each criterion has its own list of approved AWS services. You should use only the services listed for that specific criterion, and avoid using others unless you have received prior approval.

Services for general purpose are approved across the board: EC2 instances, Route53, S3, EFS, RDS, DynamoDB, CDK, SDK, CloudFormation, Parameter Store, Secrets Manager, Cognito

The page AWS Services Available provides a list of all AWS services available in the CAB432 cloud account.

Services that implement the substantially the same functionality won't count separately as there is little need to have two tools for the same job. For example, if you were to deploy your own instance of MySQL on EC2 *and* use RDS then there is no added value here; it would be better to implement both databases using the same service.

Core criteria (10 marks)

These criteria relate to core learning outcomes around cloud architecture, scalability, and security. You must attempt every core criterion.

Although there are no criteria related to deployment, it is expected that you will deploy your application to our AWS account.

Note that in this assessment the core criteria represent a smaller portion of the total grade than in the first assessment. This is because we expect that there will be a wide variety of projects, and not all of the cloud services we have studied so far will be appropriate for all projects. Hence the additional criteria, where you have the choice to attempt criteria that are most appropriate to your project, make up a larger portion than in the previous assessment.

Microservices (3 marks)

For this core criterion we require that your application has at least two separate services running on separate compute.

The separation into two services must be appropriate and not arbitrary. For example, one service might provide the main REST API or serve the web client while the other service implements the CPU intensive process that you will horizontally scale.

The two services must run on separate compute instances, eg. separate EC2 instances, separate containers on ECS, or one running on ECS and the other on EC2.

Approved services for this criterion: EC2, ECS

Load distribution (2 marks)

Your application must use an appropriate mechanism for distributing load to multiple instances of the service implementing your CPU intensive process.

An application load balancer may suitable if a single instance the service can handle multiple requests simultaneously.

A message queue accessed by multiple instances, but delivering each message to only one instance (eg. SQS) is often more suitable for processes that require more time or can only handle one task at a time.

Approved services for this criterion: Any type of load balancer, SQS

Auto scaling (3 marks)

The service handling the CPU intensive task in your application must automatically scale horizontally in response to load.

- Clients should not see interruptions in service while scaling out happens, although there may be a graceful degradation in service (eg. response times increase until new instances come online)
- The service can be deployed on ECS or EC2. It is not acceptable to use Lambda for the CPU intensive task service.
- You will need to demonstrate autoscaling from 1 instance/container up to 3 in response to load, and back down to 1 instance when load is reduced.
- The default choice for scaling metric is average CPU utilisation with the target set to 70%. It is not acceptable to reduce the target to compensate for an inability to achieve high average CPU utilisation. If you are using a custom metric then the scaling must be successful with that metric instead.

For EC2, take care that if you are using a single-threaded application that you use a single-CPU instance type (eg. `t2.micro`) and that the *Credit specification* is set to *unlimited* for `t2` instance types.

Approved services for this criterion: Auto-scaling groups (EC2), Target groups, Application Auto Scaling (ECS), CloudWatch, Lambda (for custom metric)

HTTPS (2 marks)

Your application is accessible on the public internet over HTTPS with a valid certificate. This requires:

- You have set up a subdomain of `cab432.com` in Route 53 with a CNAME record pointing to an appropriate endpoint for your application. This is not assessed here as it was assessed in A2, but it is required in order to obtain a certificate.
- You have requested and obtained a certificate using ACM.
- You are using an API Gateway or Application Load Balancer configured to use the certificate and route requests to your server instance(s) serving the publicly accessible web client and REST API.

It is acceptable to use an API Gateway or Application Load Balancer for this purpose even if you are not using other functionality that they provide. But note that in that case they don't count towards additional communication mechanism in the criteria below.

Approved services for this criterion: Route53, ALB, API Gateway, CloudFront, Certificate Manager

Additional criteria (14 marks)

We have provided an excessive number of additional criteria. You do not need to attempt all of them. Keep in mind that we will stop marking once we have considered enough additional criteria to account for 14 marks, regardless of whether you have earned the full 14 marks. There is also an "upon request" that requires approval by the unit coordinator.

You cannot achieve more than 14 marks from these tasks. We will mark only those that you explicitly tell us to consider. You should choose the most appropriate for your application and those you will achieve the best outcome for.

More details are given in the marking rubric. Be sure you are completing the tasks in such a way that the marking rubric is satisfied.

It is not expected that you can respond to all additional criteria as several of them depend on the details of your application.

Additional microservices (2 marks)

This criterion is about adding more microservices. Attempting this criteria means you will have at least four microservices in total.

The division into multiple services should be appropriate, not arbitrary. Each service should be deployed on its own compute.

Approved services for this criterion: Same as the associated core criterion

Serverless functions (2 marks)

Your application uses Lambda to deploy one or more services. Lambda must be an appropriate choice for the services being deployed in this manner. Some good examples include:

- implementing a custom mechanism for autoscaling
- responding to events, eg. queueing up processing tasks when a client has finished uploading a file to S3 via a pre-signed URL
- lightweight public-facing services

It is not acceptable to use Lambda for your application's CPU intensive task.

Approved services for this criterion: Lambda, EventBridge

Container orchestration with ECS (2 marks)

This criterion is looking for use of ECS to deploy at least one microservice.

Approved services for this criterion: ECS

Advanced container orchestration with ECS (2 marks)

In this additional criteria we will look for use of advanced ECS features. We expect at least two additional functions of: service discovery, rolling updates with failure detection, or tasks launched in response to events or on a schedule.

Approved services for this criterion: ECS

Communication mechanisms (2 marks)

Your application uses additional communication services beyond what is used for load distribution. That could include queues, API gateways, using routing functionality in an application load balancer, publication/subscription mechanisms, etc. The communication mechanisms used should be appropriate to the task.

Using API gateway style functionality in an application load balancer counts separately from its load balancing functionality. i.e., using listener rules to route traffic based on path/method/etc. counts as a separate communication mechanism.

Note that an API gateway or application load balancer used *solely* for implementing TLS does not count towards this criterion.

Approved services for this criterion: SQS, API Gateway, load balancers, EventBridge

Custom scaling metric (2 marks)

Your application must use an *appropriate* scaling metric other than average CPU utilisation to improve autoscaling.

We are looking for:

- Appropriate scaling and load distribution
    - The chosen mechanism should handle increasing traffic without overloading instances or causing service interruptions.
- Improvement over average CPU utilisation
    - The metric should better match your application's needs, leading to improvements such as faster client responses or more efficient cloud resource use.
- Scalability across different sizes
- The mechanism should work effectively whether your application runs on a single instance or scales up to hundreds of instances.

Note: "Scaling mechanism" is used broadly here. You may use a target metric, simple scaling, or step scaling.

Approved services for this criterion: CloudWatch, Lambda

Infrastructure as code (2 marks)

For this criterion you should aim to use IaC to deploy the AWS services that your application uses relating to core and additional criteria for this assessment.

We won't evaluate deployment for services covered in Assessment 1 or 2. Basically, this means that services listed under Block 1 or Block 2 on AWS Services Available.

Approved services for this criterion: Terraform, CDK, CloudFormation

For other technologies, please ask the teaching team.

Dead letter queue (2 marks)

For this criterion, you should take advantage of the dead letter queue (DLQ) feature of SQS to appropriately handle messages that cannot be processed successfully. It only makes sense to attempt this criterion if you are already using SQS. The intention is to redirect messages that repeatedly fail processing to the DLQ instead of being lost or endlessly retried. You must implement appropriate handling of the messages that end up in the DLQ.

Approved services for this criterion: SQS and associated workers that consume messages on the DLQ

Edge caching (2 marks)

Your application should make *appropriate* use of edge caching with Cloud Front. This means:

- You should have a convincing reason that the data you are caching will be accessed frequently. This does not have to be true *now* but it should be true in an imagined wide-scale deployment of your application.
- The data that you are caching needs to be infrequently changed. For our purposes this basically means that it should be static. Resources such as static front end HTML/CSS/JS/image files (including those built with React and similar) are good candidates.

Approved services for this criterion: CloudFront, ElastiCache

Upon request (2 marks)

This additional criteria exists once in the rubric. You must seek approval from a coordinator.

This criteria gives you the opportunity to gain marks for other functionality or aspects that demonstrate a high level of achievement in the project. Be sure to **first** speak to a coordinator if there is something specific that you'd like to do and get additional credit for.

Write a report (21 marks)

Your report will cover the following topics:

- A description and diagram of the architecture of your application
- Justification for your architecture choices in the design of your application
- A discussion of further development that would be necessary to make your application scalable and secure for a large-scale deployment
- Implications of your architecture choices on sustainability
- Inclusion of a cost estimate, calculated in the **AWS Pricing Calculator** **(https://calculator.aws/#/)**

Further details on the report can be found at **3.1 Submission specification** **(https://canvas.qut.edu.au/courses/20367/pages/3-dot-1-submission-specification)**

Technologies and special permission

We will follow the same guidelines as for Assessments 1 and 2. The following technologies do not require special permission:

- Technologies that you already have permission for from assessment 1 or 2
- All AWS services listed in AWS services available.

Submission

Your submission will have three parts:

- Your code
- A video demonstrating the functionality of your application, similar to those from Assessments 1 and 2
- A report

Further details on submissions can be found at **3.1 Submission specification** **(https://canvas.qut.edu.au/courses/20367/pages/3-dot-1-submission-specification)**

Feedback

Under normal circumstances, you will receive marks for each criterion via a Canvas rubric within 10-15 working days of submission. Click on Marks to see your results. Usually the reason for each choice of mark is self-evident, the marker will include some written feedback about your performance. You should use this feedback to strengthen your performance in the next assessment item.

Moderation

All staff who are assessing your work meet to discuss and compare their judgements before marks or grades are finalised.