Link between what you research, and the work delivered.

Referencing

* Learn about referencing in-depth
* Primary and secondary research
* IEEE (Peer review = high quality)
  + IEEE
  + Journal 1
  + Journal 2
  + Etc

# Example:

* Devops improves the amount of bugs etc
* Can you find anything that suggests this is wrong?

Discover = uni library

Solving the problem in this way

Research previously

Literature review

* Research question to attempt to answer that question

PR

* Does the application hit the PR KSB’s?
* Ci/cd pipeline that already exists
  + Prove it
  + It improves this this and this
  + Software engineering, improving a process
    - Have you hit the KSB’s

Read the abstract

* No more than 1 paragraph

Literature review now to understand and formulate a research question

Concept maps (mindmaps)

* Use this

Google scholar

* Keywords etc

Reading  
- review abstract and conclusion

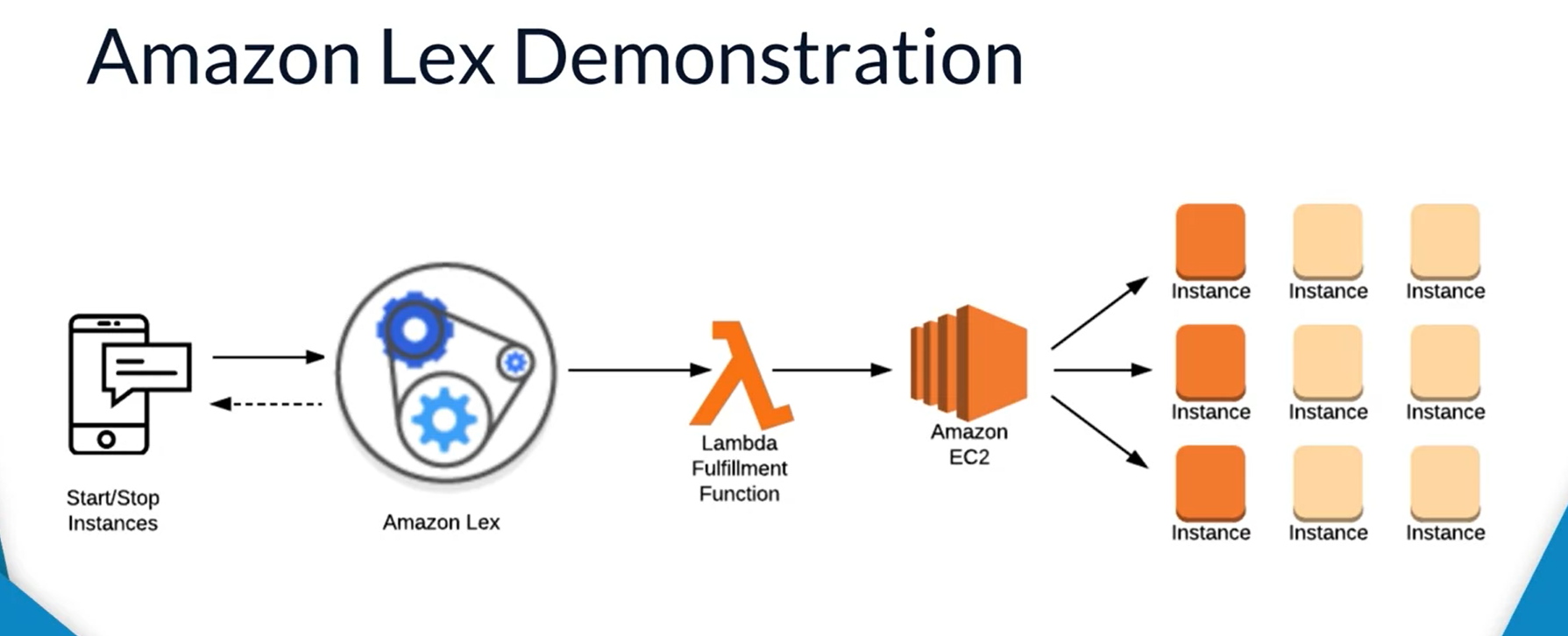
Make a note of the source

Create a table for the bibliography of citations

* Zotero

Broad and deep literature review

Contradictory literature against the question also.



KSB’s

We’re running this in an agile way, so we don’t know what the method is, possibly agile.

* Gitlab
  + ci/cd tickets

professionally present digital and technology solution specialism plans and solutions in a well-structured business report;

**architect build and support leading edge concurrent software**

* test coverage
  + unit tests or other tests
* industry standards
  + performance
  + times
  + availability
  + scalability
  + redundancy
  + cloud\*

**drive the technology decision making**

* discussions with client
  + you’re driving that
* considering current technologies
  + argue, I will not use them due to
* evaluation different technology design
  + I did consider other options on how to implement and design it
    - Azure/aws
    - Hybrid
    - Other programs etc

Develop and deliver distributed

* Doesn’t have to be distributed or complex
  + Simple
* In the cloud its distributed already
* Scalable? Of course in the cloud

Innovative user experience

* My example of the app would be good.

Update current software products

* Sql scripts can be updated to become easier

Planning tasks

* Jira board etc
* Security guidelines
* Code quality guidelines

Accountable quality deliverables from one or more dev teams

* Evidence that through the client

Various inputs

* Email meetings
* Sprint retrospectives
* Legal guidelines
* Governance
* Guided the architecture that you’re creating

Methodologies designed to help create

* Agile
* Waterfall etc

The approaches used to moduliarise

* The way python is packed (modularised)
* They way you modularise code

How to design and deploy

* Security
* Requirements of the stakeholder
* Successful code, well written and readbale