# Migrating to Azure

Migrating to Azure

- Many organizations want to migrate their apps to the cloud

- Various motivations

- Has to be planned ahead, and done methodically

- Rushing to the cloud could end up badly

- There's a well defined process that should be followed in order

  for the migration to complete successfully

The Migration Process

Motivation Assessment

System Assessment

Migration

Enhancements

Testing

Go Live
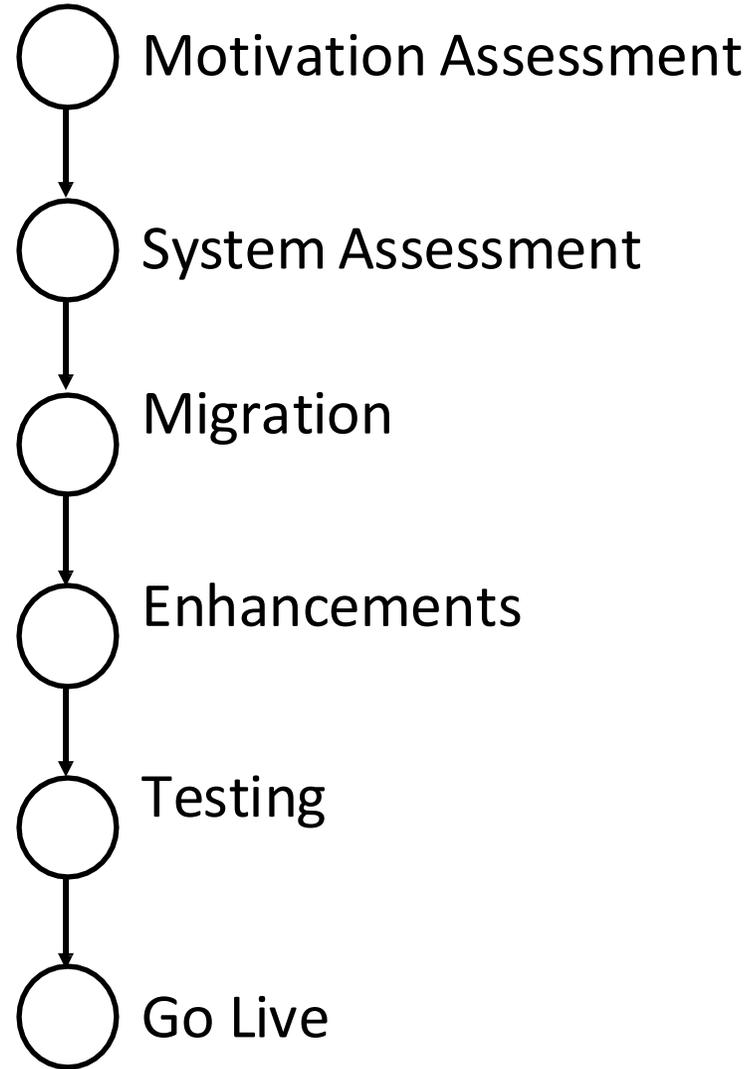
Motivation Assessment

- Answer the grand question:

> ## Why?
> (do we want to migrate to the cloud)

Possible Answers

- To save costs

- To take advantage of cloud capabilities (services, redundancy, scalability, etc)

- To modernize the system

- To attract new employees

Saving Costs

- Not always the case

- Sometimes cloud system can cost MORE than on-premises system

- CapEx vs OpEx

- A very thorough cost estimation should be conducted

Take Advantage of Cloud Capabilities

- Great motivation! (probably the best…)

- Make sure the migrated system can actually use cloud capabilities

- Example: Legacy, single-session app that cannot be scaled will

  end up on a single VM, just like on-prem

Modernize the System

- Usually, Modernization = Rewrite

- Why?

- Possible answers:

  - Not supported platforms

  - Difficult to hire employees

  - Licensing costs

Attract New Employees

- Not a very good motivation, but it works…

- The organization has to be part of the cloud trend in order to

  attract new employees

Migration Strategies

Lift & Shift →	Install the app in the cloud "as-is", without any changes, usually on a VM
Refactoring →	Modify the app a bit so it can run on managed services (ie. App Services)
Rewrite →	Rewrite the app from scratch, usually in a modern platform, and take advantage of all cloud services

Lift & Shift

On-Premise

VM
Java App

VM
Oracle DB

Cloud

Refactor
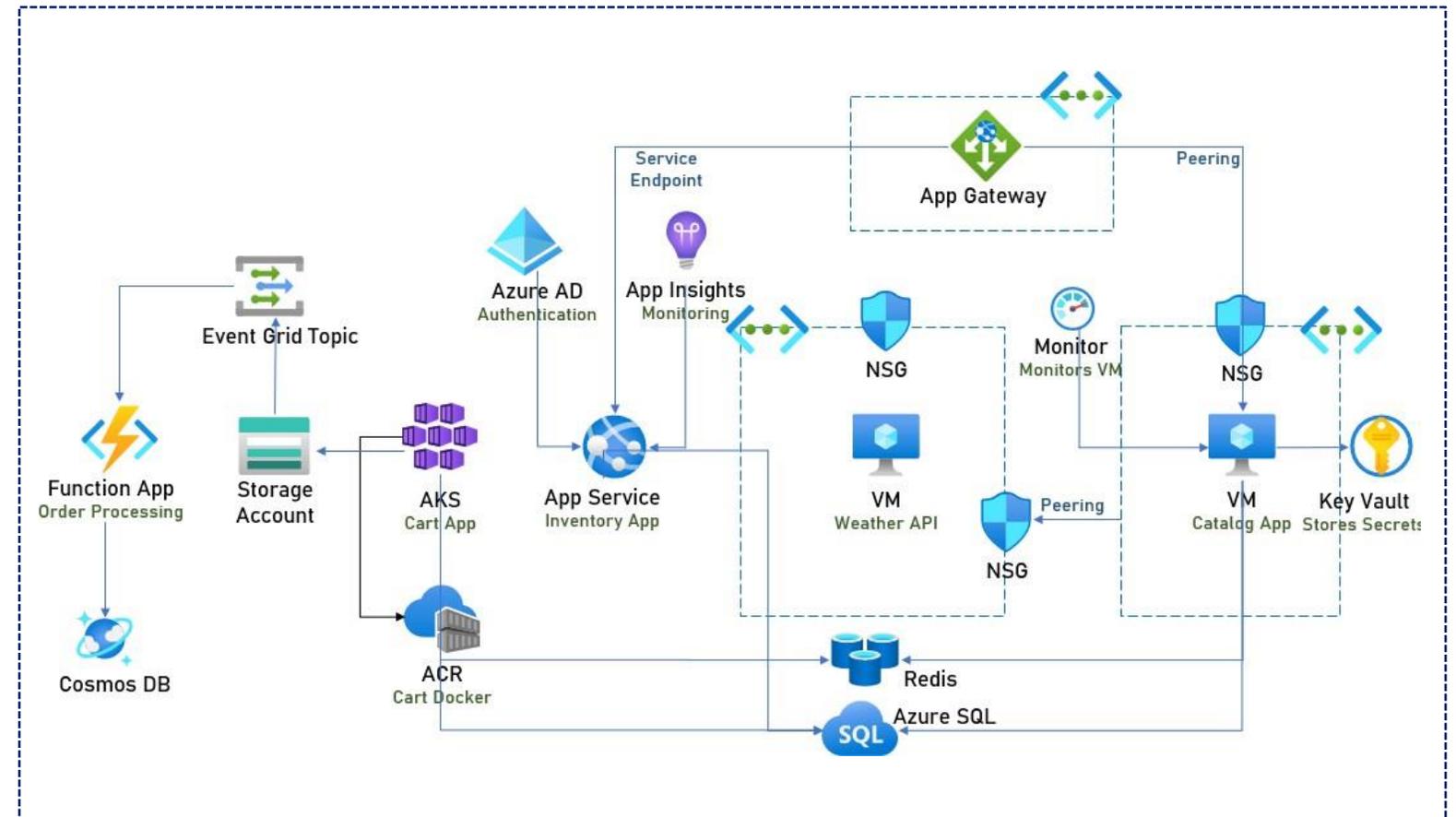
Rewrite
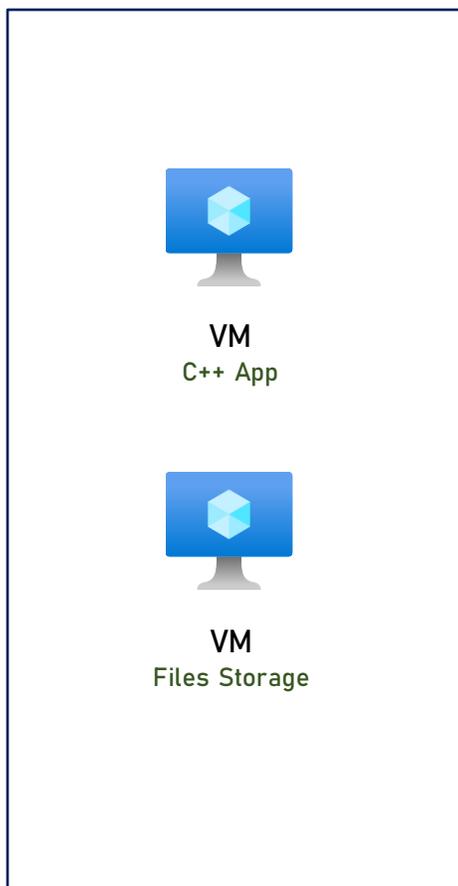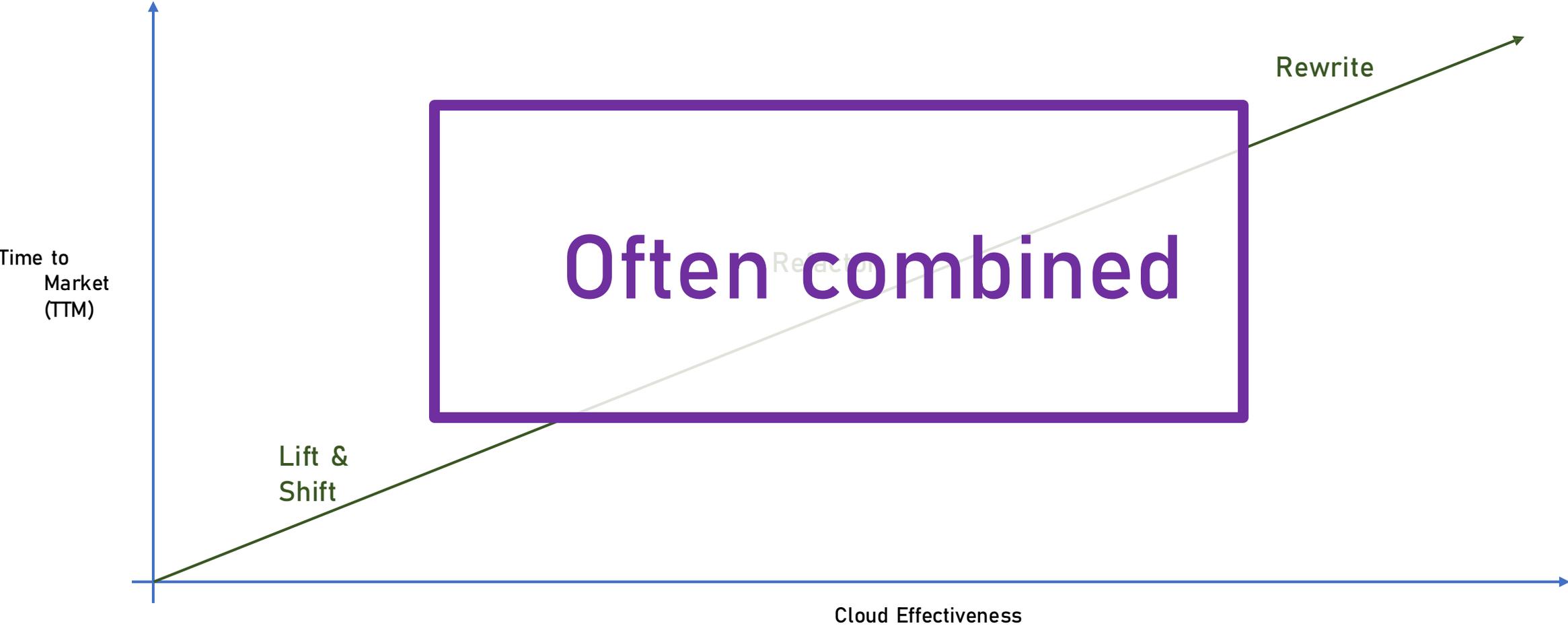
Comparing Cloud Migration Strategies

System Assessment

- Goal:

  - To find out what is the best way to migrate the system to

    the cloud

  - Doing this by asking the right questions

System Assessment

- Questions we need to ask:

What is the runtime platform of the system and on which OS? (.NET, Java, C++ etc.)

What is the authentication mechanism?

How is it deployed today? (VM, Docker…)

Any interfaces to organizational

What database(s) is used?

Any work with hardware? (dongles…)

System Assessment

- Answers dictate migration strategy, if possible at all:

What is the runtime platform of the system and on which OS? (.NET, Java, C++ etc.)

→ What is the authentication mechanism?

How is it deployed today? (VM, Docker…)

Any interfaces to organizational

What database(s) is used?

Any work with hardware? (dongles…)

System Assessment

- Answers dictate migration strategy, if possible at all:

What is the runtime platform of the system and on which OS? (.NET, Java, C++ etc.)

What is the authentication mechanism?

How is it deployed today? (VM, Docker...)

Any interfaces to organizational

What database(s) is used?

Any work with hardware? (dongles...)

System Assessment

- Answers dictate migration strategy, if possible at all:

What is the runtime platform of the system and on which OS? (.NET, Java, C++ etc.)	What is the authentication mechanism?
How is it deployed today? (VM, Docker…)	Any interfaces to organizational
What database(s) is used?	Any work with hardware? (dongles…)

Azure Migrate

- A central hub for assessing and migrating various resources

- Uses platform-specific agents to discover the resources that can be migrated

- Provides migration recommendations

- Free

Azure Migrate

Azure Migrate

- Good for low-level assessment (platforms used etc.) but won't deal

  with other considerations

  - ie. Interfaces with organizational systems

- Can be used as a first-step, but must perform manual

  assessment afterwards

Migration

- Time for actual migration…

- Migration strategy depends on the system assessment results

Lift & Shift

- The easiest strategy

- Simply create VM images and build new VMs in the cloud

- Don't forget to put VMs in appropriate VNet and define NSG

Refactor

- Should be done in two phases:

  - Lift & Shift – Make sure the app works in the cloud as-is

  - Refactor – gradually improve the system and integrate

    cloud services

- Don't try shortcuts!

Rewrite

- Not really a migration…

- Design the system from the ground up to be cloud native

- Use as many cloud services as possible

DB Migration

- In general:

  - Prefer the managed version of the DB (and not on a VM)

  - Migration is better done using the DB's own tools and not via
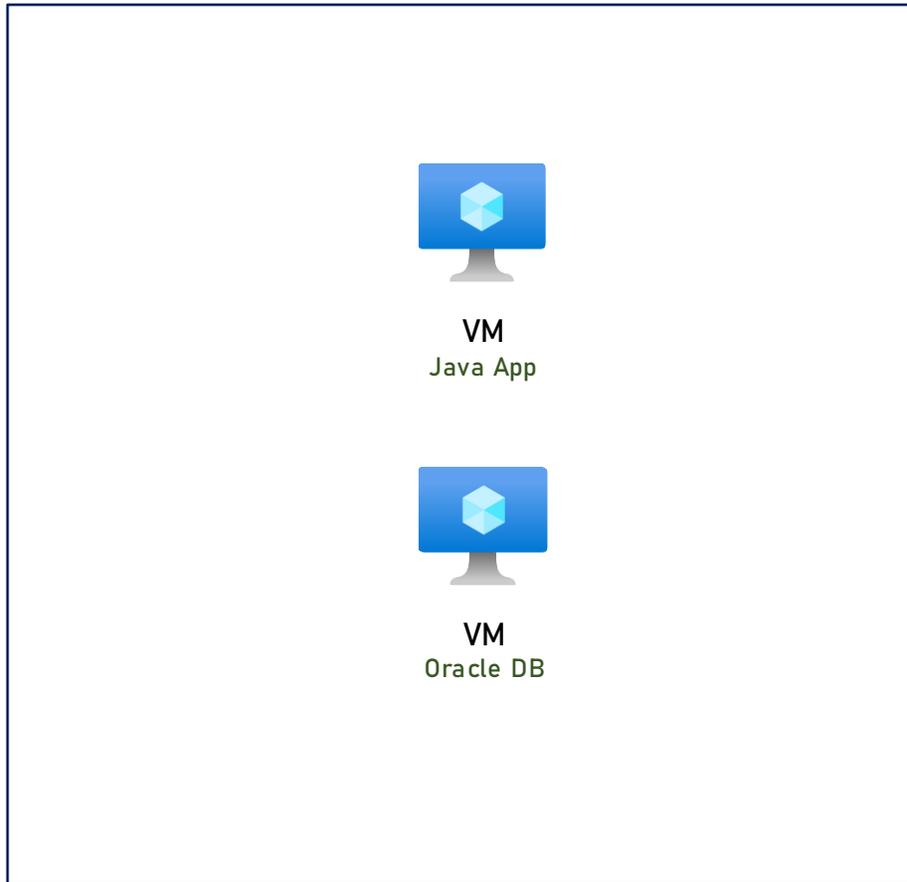
    Azure migration services
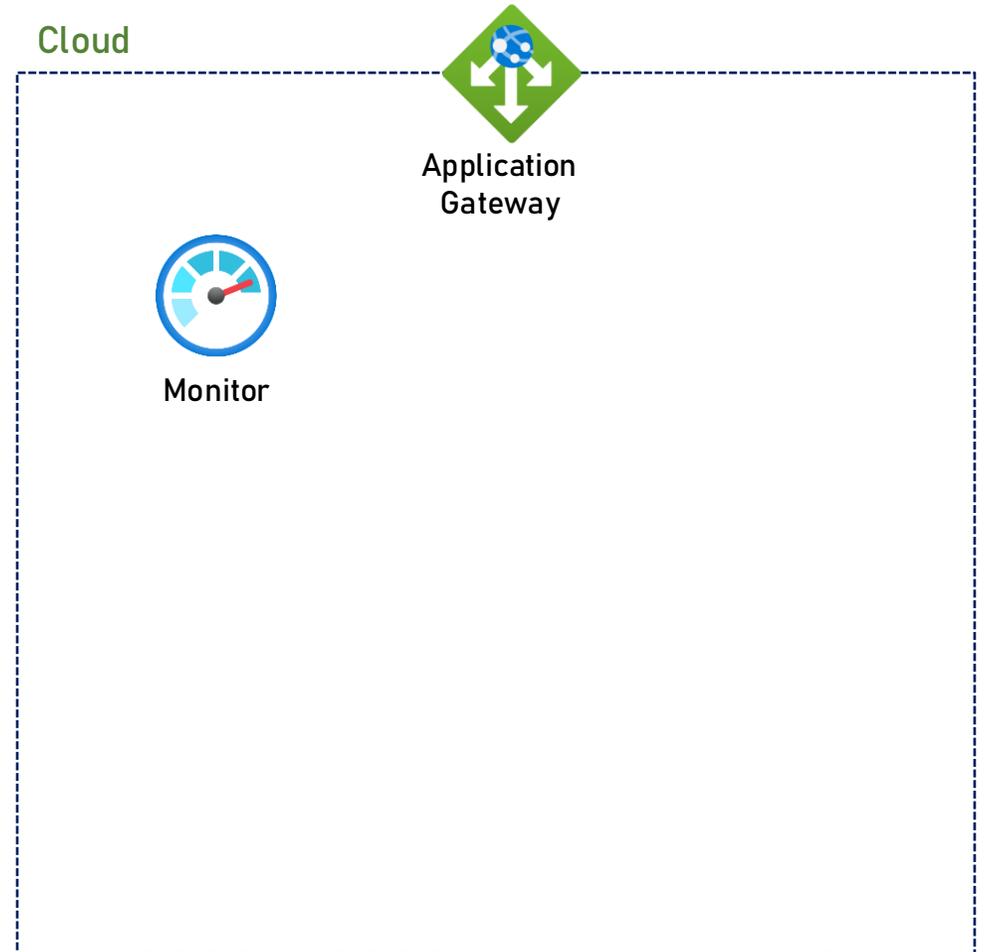
App Enhancements

- After migration, enhance the system

- Not necessarily requires code changes

- Main areas for improvements:

  - Logging and Monitoring using Azure Monitor

  - Network protection using Application Gateway

Lift & Shift Example

On-Premise

VM
Java App

VM
Oracle DB

Cloud

Application
Gateway

Monitor

Testing

- Testing in cloud is similar to on-prem

- Put strong emphasis on logging and monitoring

- Make sure you have access to system's data

- If system is auto-scaled or DRed – test these scenarios

- Check performance – might vary from the on-prem figures

Go Live

- Congratulations!

- Keep an eye on the costs

  - Set up budget alerts

  - Define tags

  - Look at the data at least once a month