Essential Cloud Technologies

Contents

Cloud Fundamentals

</conygre training

Objectives

- Introduction to the Cloud
- Benefits of the Cloud
- Strategies for Migrating to the Cloud

</conygre training

What is '*The Cloud*'?

Why Not Use Your Own Computer?

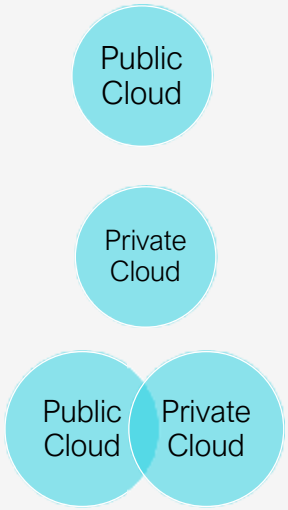
1. Save capital expense and swap with variable expense
2. Benefits of the economies of scale of the cloud provider
3. Flexibility and Elasticity
4. Speed and agility
   - How long does it take you to get a server in your data centre?
5. Let a company who's 'day job' is running data centres run yours so your company can focus on their 'day job' which probably isn't running data centres

Additional Services

- Elastic load balancing
- Serverless databases, compute, storage
- Easy and automatic replication across datacentres
- Cheap and reliable high availability
- Cheap and reliable file storage

Cloud Terminology

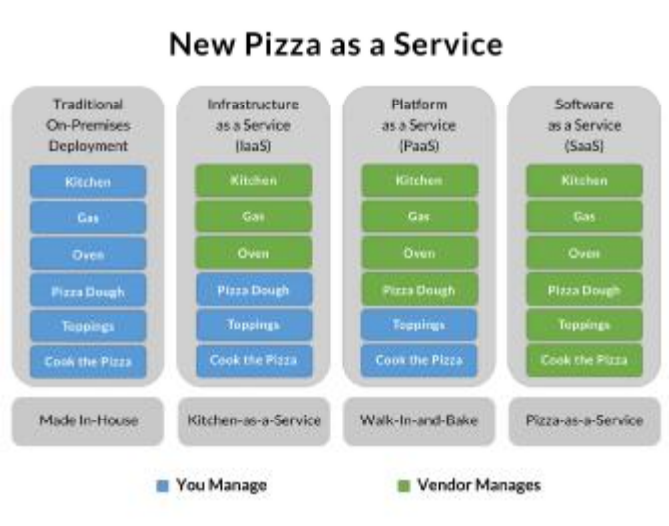
- Public Cloud
  - Use a cloud provider ie. 'somebody elses computer'
- Private Cloud
  - Implement some basic cloud capabilities in our own data centres
  - Usually virtual machines and containerisation
- Hybrid Cloud
  - Do a bit of public and private and link them together



X as a Service

- Some people use these terms when describing cloud technologies
  - Infrastructure as a Service
  - Platform as a Service
  - Software as a Service
- This terminology can be confusing - so let's use an analogy!

Pizza as a Service!



https://m.oursky.com/saas-paas-and-iaas-explained-in-one-graphic-d56c3e6f4606

Approaches to Migration

- There are several approaches that companies take to implementing the cloud
  - All in and Forklift
  - All in and Leverage
  - Hybrid Architecture

All in and Forklift

- Move wholesale to the cloud
- Take applications running on premises and move them to the cloud as is
  - Hence the term forklift or sometimes called *lift and shift*
- Pros
  - Relatively easy to do
  - Relatively quick to do
  - Not much cloud knowledge required
- Cons
  - You are not benefiting fully from the capabilities of what cloud provision can achieve

All in and Leverage

- Take applications that are not in the cloud and re-architect them to take advantage of cloud
- Pros
  - Benefit from the advantages of the cloud platform
- Cons
  - More complicated
  - More up front expense

Hybrid

- Hybrid is a mix of on premises and cloud
- Some applications and data cannot go to cloud
  - Regulatory compliance
  - Security requirements
- Many applications can move to the cloud
  - Make sure you have a fast and secure connection between on premises and cloud provider
- Hybrid is where most large corporates are right now

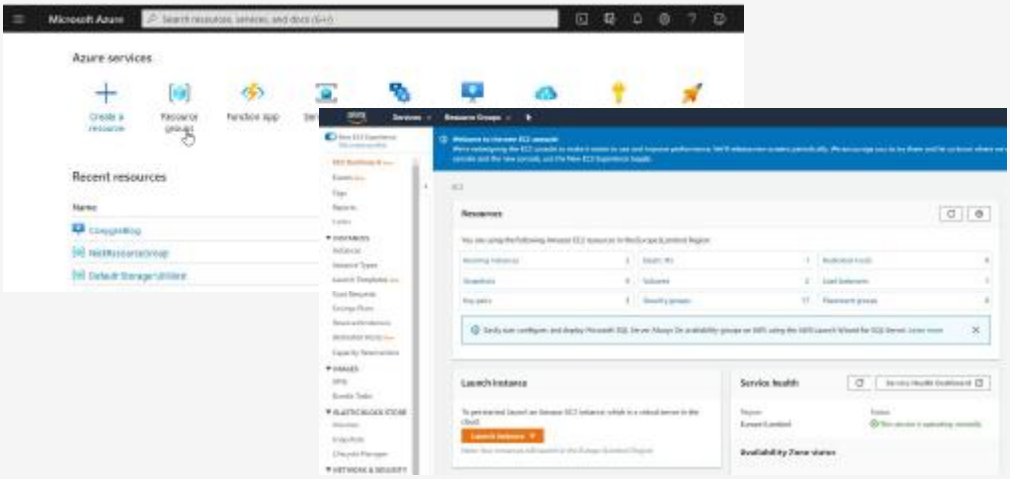
Public Cloud Providers - the Main players

- Amazon Web Services (AWS)
  - The market leader and pioneer of cloud
- Microsoft Azure
  - The second largest and MS have the benefit that nearly every organisation in the world is already a Microsoft customer!
- Google Cloud
  - Many would say is catching up with AWS and Azure

Interacting with the Public Cloud

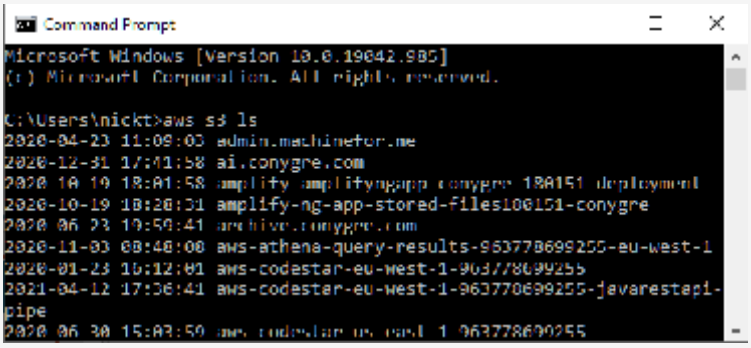
- Cloud providers offer three main ways of working with the platform
  - Web consoles
  - Command line tools
  - Programming Code

Web Consoles



Command Line Tools

- Anything you can do with the Web consoles can also be done via commands that can be executed in PowerShell or Shell Scripts or at Command Prompts



Programming APIs

- The Cloud providers also have created programming APIs that can be used by developers to work with the cloud platform

```
const AWSXRay = require('aws-xray-sdk');
const XRayExpress = AWSXRay.express;
const express = require('express');

 / Capture all AWS clients we create
const AWS = AWSXRay.captureAWS(require('aws-sdk'));
AWS.config.update({region: process.env.DEFAULT_AWS_REGION  | 'us-west-2'});

 / Capture all outgoing https requests
AWSXRay.captureHTTPsGlobal(require('https'));
const https = require('https');
```


Interaction with the Cloud Best Practices

- Although you have all these options, using the Web console and allowing humans to do things is generally the worst way to do it
- You get fat finger errors!

WHOOPS: A Fat Finger Error Sent A Stock Shooting Up $319 In Less Than A Second This Morning

- Using automation is always preferred!

Summary

- Introduction to the Cloud
- Benefits of the Cloud
- Strategies for Migrating to the Cloud

Cloud Architecture Overview

Objectives

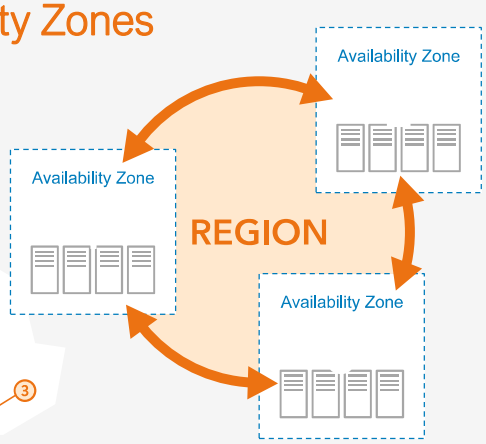
- Regions
- Availability Zones
- Services

Regions - AWS

Regions - Azure

Availability Zones - Azure and AWS

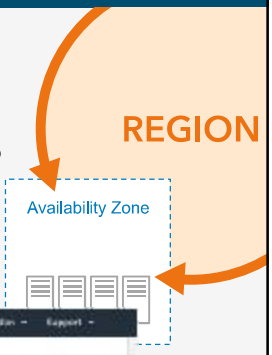
- Each Region is made up of Availability Zones (between 2 and 6 of them)
- An Availability Zone is essentially a data centre or cluster of centres
  - Connected by a high bandwidth, low latency, encrypted data connection

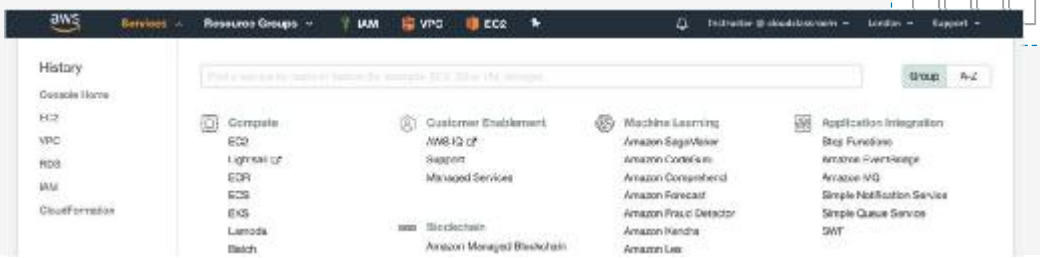


Services

- Cloud providers then provides Services
- For example AWS launched in 2006 with 3 services
  - Today there are over 212 to choose from*

*as of February 2020





Services Examples

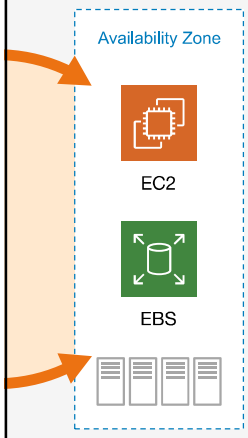
Capability	AWS	Azure	Google Cloud
Running Servers	EC2	Virtual Machines	Compute Engine
Storage	S3	Azure Blob Storage	Google Cloud Storage
Databases	RDS / DynamoDB	SQL Managed Instances	Cloud SQL
Containers	Elastic Container Service	Azure Container Service	Google Kubernetes Engine

- We will explore some of these services now using AWS as our example platform
  - Remember the principles are the same on all the platforms but we had to pick one so as to not make this course too confusing!

Running Servers

- Regardless of cloud provider, one of the fundamental capabilities is that of running virtual servers in the cloud
  - EC2 in AWS
  - Virtual Machines in Azure
  - Compute Engine in Google Cloud

Services — Elastic Compute Cloud (EC2)



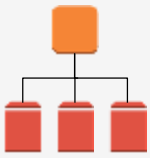
- EC2 is Amazon's virtual server service
- An EC2 server is called an instance
- Each instance is launched from an Amazon Machine Image (AMI)
  - The template for the machine
- Each instance is launched with an instance type
  - The size of the machine eg. memory, CPU etc.

AMIs

- Servers are launched into availability zones from Amazon Machines Images (AMI)
- AMIs can be
  - Standard Linux and Windows variations
  - Third Party Provided – lots to choose from!
    - AWS Marketplace
    - Azure Marketplace
    - Google Marketplace
  - Created by you

Hard Drives

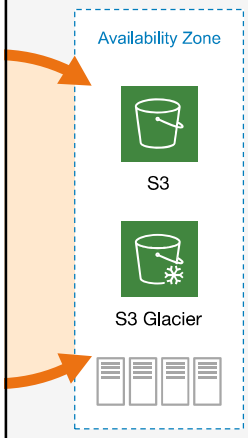
- EC2 servers use Elastic Block Store drives (EBS)
- EBS volumes can be
  - Magnetic
  - Solid State
  - Provisioned with a minimum read / write speed
- EC2 instances can have multiple EBS drives
- Drives can be moved between instances
  - Just like physical drives



Data Storage

- On all cloud platforms there are many storage options
  - Sometimes referred to as polyglot persistence!!
- The AWS options include
  - EBS – hard drives
  - Instance store – temporary SSD storage physically attached to EC2 instances
  - Elasticache – Memcached / Reddis as a service
  - DynamoDB – NoSQL database as a service
  - RDS – relational database as a service
  - S3 – unlimited object storage
  - Glacier – archive service

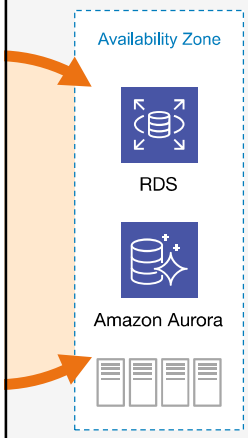
Services — Simple Storage Service (S3)



- S3 is a file storage service over HTTP/S
- Files are called objects
- Objects are stored in buckets
- Highly available — 99.9% availability
- Highly durable — 99.999999999% durability

- This is called Blob storage on Azure

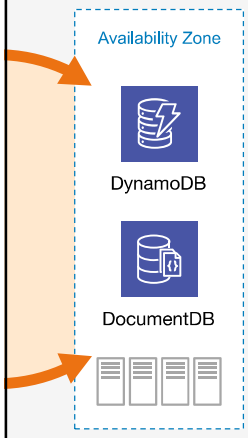
Services — Relational Database Service (RDS)



- RDS is a managed relational database service
  - Support for Oracle, Microsoft SQL Server, MySQL, MariaDB and PostgreSQL
- Simple configuration of **clusters** and **backup**
- Amazon's own database is called Aurora
  - Compatible with MySQL and PostgreSQL

- This is called SQL Managed Instances on Azure

Services — DynamoDB



- DynamoDB is a NoSQL database service
- Targeted at high performing applications
  - Single digit millisecond latency at any scale
- Highly available — 99.999% availability
- There are other NoSQL databases services such as
  - DocumentDB – 'MongoDB as a service'
  - Neptune – Graph based data store

Azure has a service called Cosmos that is very similar to DynamoDB

Serverless Architectures

- Removing the need for servers altogether is AWS Lambda or Azure Functions
  - These services allow you to deploy functions to the cloud written in multiple programming languages
- Functions can be triggered by
  - REST API calls
  - Messaging
  - File uploads
  - Scheduled events
  - Voice commands … and more

Serverless Function Benefits

- You only pay for the time used when the function is invoked
- There is no infrastructure to support
- You can specify the amount of memory the lambda runs with
- Lambdas can be written in
  - Java
  - JavaScript
  - Python
  - C#
  - Go
  - Ruby

Summary

- Regions
- Availability Zones
- Services

Cloud Security

</conygre training

Objectives

- Security concerns
- Securing your Account
- Security your infrastructure
- Security your data
- Encryption
- Auditing tools

</conygre training

Security Concerns

- Securing your cloud accounts
- Securing your infrastructure
- Securing your data

conygre training

Securing your Account - Azure

- Azure Security is based around identities which are then managed in Azure Active Directory
- Identities are then given permissions to access different services within your account through Roles
- Active Directory is a product that many Microsoft customers will already be familiar with as a way of managing users, and conveniently, this is the same thing that is used for cloud accounts

conygre training

Securing Accounts

- When you set up an AWS account, you will have a username and password
  - This should *RARELY* be used
  - Keep it under lock and key
- Instead of using your root account, AWS provides
  - Identity and Access Management (IAM)

</conygre> training

Understanding IAM

- IAM allows you to create
  - Users
  - Groups
  - Roles
  - Policies


Users


Groups


Policiies


Roles

</conygre> training

Securing Servers - YOU

- YOU are responsible for
  - Patching and updating your servers
  - Malware protection and antivirus
  - Ensuring that you only open the appropriate ports through appropriate security groups
  - Ensuring that you place servers in appropriate subnets with the correct routing rules
- Cloud providers do provide tooling to assist you, but it is YOUR responsibility to use them

Securing Servers - The Cloud Provider

- The Cloud Provider is responsible for
  - All your security groups and routing rules work as you have specified
  - EC2 instances and virtual machines are isolated from each other
  - The security of the physical data centres
  - That all disks at end of life are destroyed to the appropriate international standards

Securing Data

- You are responsible for things like
  - For putting the appropriate permissions and access rights onto your S3 / blob storage, DynamoDB / Cosmos tables, databases etc.
- The Cloud Provider is responsible for things like
  - Ensuring the permissions you have set actually work
  - Backups and patching of SQL databases happens according to your schedule

Encrypting Data

- Many data services support encryption for example, blob storage and the hard drives of virtual machines can be encrypted at rest with keys
  - Provided by the cloud provider
  - Provided by you

Tracking Cloud Account User Actions

- Cloud Providers give the ability to track your user actions
  - CloudTrail on AWS
  - Activity Log on Azure
- These services tell you
  - Who does what and when
  - Who stopped that server?
  - Who created that queue?
- All data can be permanently stored
- Great for audit trails


SAAS

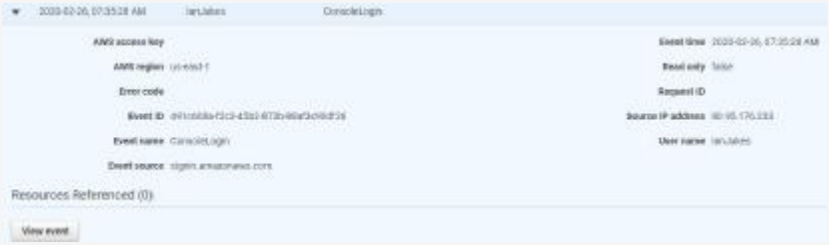
Vengeful sacked IT bod destroyed ex-employer's AWS cloud accounts. Now he'll spent rest of 2019 in the clink

Bloke hit delete on £500,000 of 'business-critical data' after he was let go for 'poor' performance

https://www.theregister.com/2019/03/20/steffan_needham_aws_rampage_prison_sentence_voova/

Cloud Trail Example

- The information includes things like
  - User name
  - Source IP
  - Time / Date
  - What they did

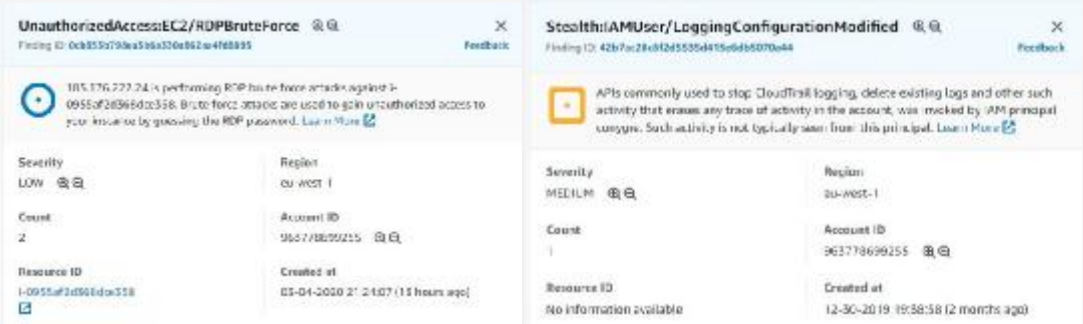


Advanced Threat Detection

- Public cloud providers now provide AI driven threat detection services to their platforms
  - Azure Advanced Threat Protection (Databases only)
  - AWS Guard Duty
  - Google Cloud Armor

AWS GuardDuty

- The Guard Duty service uses AI to monitor what is going on in your AWS account in real time and highlight anything unusual or concerning



Web Firewalls

- Web firewalls allows you to create firewall rules to block traffic to your Web resources
- These rules can be managed across accounts or just within a single account
- Rules can be used to block things like
  - IP addresses
  - Geolocations
  - SQL injection
  - URL patterns

Information Protection

- Amazon Macie allows you to identify where you are storing personal identifiable data
- It provides insights into where this data is within S3 and how it is being accessed
- It will highlight any publicly visible data

- No equivalent on Azure at this time

Security Best Practices

- The public cloud providers offer all sorts of recommendations and best practices around security
- The AWS Five Pillars also highlight a number of questions you can use to critically look at your security
  - https://aws.amazon.com/blogs/apn/the-5-pillars-of-the-aws-well-architected-framework/
- On Azure you can use the Best Practices guide for security
  - https://docs.microsoft.com/en-gb/azure/security/fundamentals/best-practices-and-patterns
- Google provide the Security Foundations Guide
  - https://services.google.com/fh/files/misc/google-cloud-security-foundations-guide.pdf

Summary

- Security concerns
- Securing your Account
- Security your infrastructure
- Security your data
- Encryption
- Auditing tools

Storage Options

conygre training

Objectives

- Overview of Storage Options
- Object Storage
- Relational Databases
- Big Data
- NoSQL Storage

conygre training

Overview

- There are multiple data options in the cloud
  - Relational Database options
  - NoSQL options
  - Object Storage
  - Disk storage

Relational Databases

- When using the Cloud platforms, there are services that will address the following requirements for you
  - Installation
  - Patching
  - Backups
  - Master / Backup configuration
  - Read Replica configuration (on some databases)
  - Complexities around resizing a live database instance

Supported Databases

- The cloud platforms support all the commonly used database products including
  - Oracle
  - SQL Server
  - MySQL
  - Postgres

conygre training

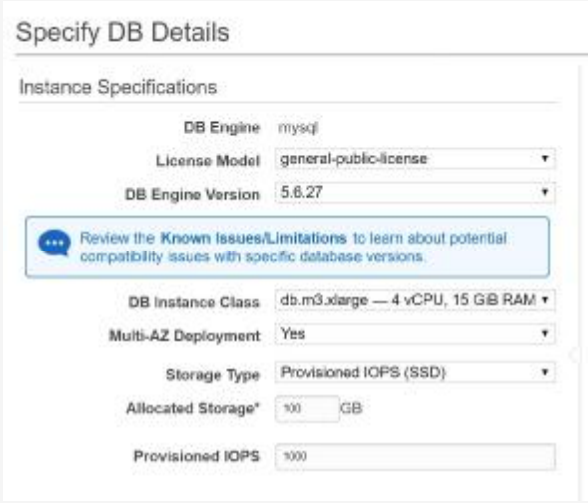
Relational Database Service Considerations

- When using these database services you can choose
  - Different instance sizes
  - Backup time
  - Patching time
  - Availability Zone locations for Master/Backup
  - Encryption at rest and or in transit
  - How Access is managed

conygre training

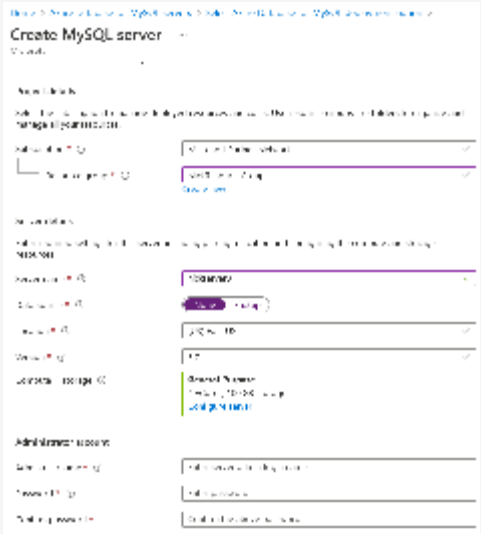
RDS Example on AWS

- RDS instances can be configured using the Web console
- Notice that setting up a master / backup in different AZs is done using just a drop down box!



SQL Database on Azure Example

- Here is a screen shot of the Azure portal being used for a MySQL instance



Data Warehousing

- On Azure, you can use Azure Synapse
- On AWS, you can use RedShift

- These tools can be used for petabyte scale relational data
- They can be queried using SQL with JDBC/ODBC drivers to gain information from your data

NoSQL Options

- The cloud providers offer multiple NoSQL Database options
  - DynamoDB / Cosmos
    - AWS / Azure Serverless NoSQL database
  - DocumentDB / MongoDB Cloud
    - MongoDB as a service
  - Managed Cassandra Service
  - ElasticSearch Service

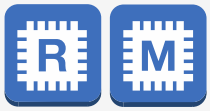
- These services can offer very fast and reliable storage with some at any scale

NoSQL with DynamoDB

- For NoSQL data, AWS provides DynamoDB
- According to AWS
  - *"Amazon DynamoDB is a fast and flexible NoSQL database service for all applications that need consistent, single-digit millisecond latency at any scale"*
- You pay for the provisioning of throughput and the storage
- DynamoDB supports key based and document based storage

In Memory Storage

- On AWS, Elasticache is a managed cache service supporting two common cache platforms
  - Memcached
  - Redis
- You can use it to store petabytes of data in memory through clusters of cache nodes



- On Azure, the equivalent is Azure Cache for Redis

Object Storage

- For Object Storage,
  - S3 / Blob Storage
    - General purpose storage
    - Immediately accessible
    - Can be made available online
  - Glacier / Archive Blob Storage
    - Archive storage
    - Not immediately accessible
    - Cannot be made available online
    - The cheapest storage option of all

S3

- S3 provides object storage for files of up to 5TB in size
- The storage is unlimited
- High durability (99.9999999%)
  - 11 nines of durability achieved by
    - Files saved on multiple drives across multiple data centres
    - *"If you store 10,000 objects with us, on average we may lose one of them every 10 million years or so. This storage is designed in such a way that we can sustain the concurrent loss of data in two separate storage facilities."*

S3 Buckets

- When using S3 you create buckets into which you place objects
- These objects have a key which is the object identifier
- Buckets can be made available over the Internet
  - Static Web sites can be hosted from buckets
  - CSS / JS / Images can be hosted from buckets

S3 Features

- Versioning
  - If enabled, all previous versions of your objects are maintained
- Encryption
  - This can be at rest and or in transit
- Public access
  - Can be enabled / disabled by you on a per object basis

Uploading options

- Data can be uploaded
  - Over the Internet
  - Over Direct Connect
  - Uploaded using SnowBall
  - Uploaded using SnowMobile

Glacier

- Glacier is the data archiving service
- It is ideal for data that you have to store but will access infrequently or never
  - Historical logs
  - Archives
  - Old Backups
  - Regulatory compliance
- Glacier is similar to S3 with 11 9's of durability
- S3 objects can be put into a lifecycle automating a transition to Glacier

Summary

- Overview of Storage Options
- Object Storage
- Relational Databases
- Big Data
- NoSQL Storage

Running Networks and Servers

</conygre training

Objectives

- Setting up servers
- Server lifecycle
- Setting up the network

</conygre training

Launching Servers

- Launching servers regardless of cloud provider require you to specify options such as
  - The machine image to use
  - The size of the machine
  - The network you want it to run in
  - What ports you would like to expose

conygre training

Machine Images

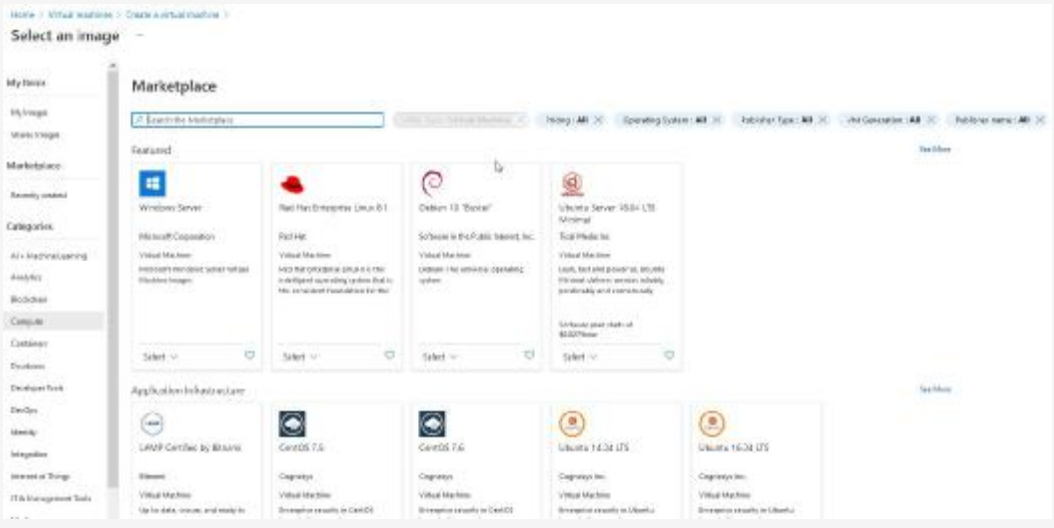
- The Machine Image is the baseline install for your server
- The image can be
  - Windows or Linux
  - Provided by your cloud provider, you, or a third party
  - Can include your entire application, or just be a basic Windows / Linux operating system
- The image is the template for your server
  - Like a VMWare Appliance or a Virtual Box image

conygre training

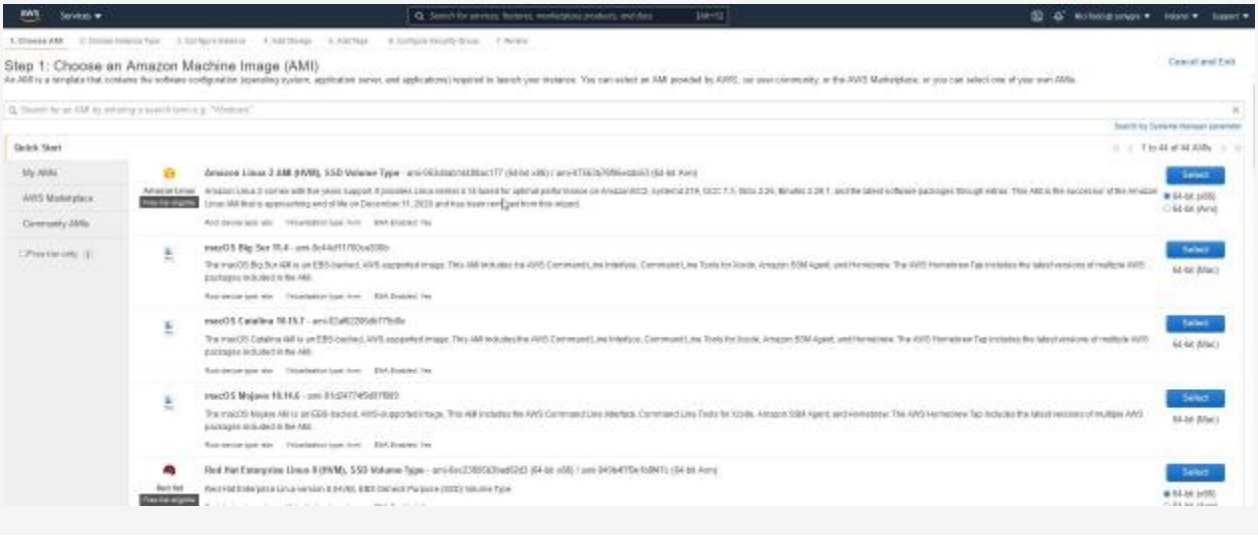
Sources of Images

- The Cloud provider will provide 'stock images' for standard Linux, Windows etc
- Marketplaces
  - Where vendors can sell pre-packaged images with applications preinstalled, for example
    - Tomcat
    - Jenkins
- Your own images
- Images shared with you from another cloud account

Azure Images



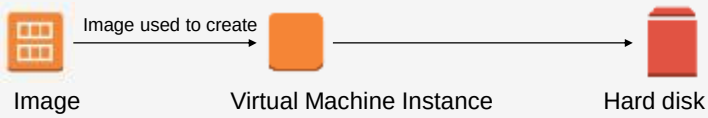
Amazon Images



conygre training

Virtual Machine Instances

- The instances are the servers that you run based on the image
- An instance will have at least one hard drive and will be of a certain type that you decide



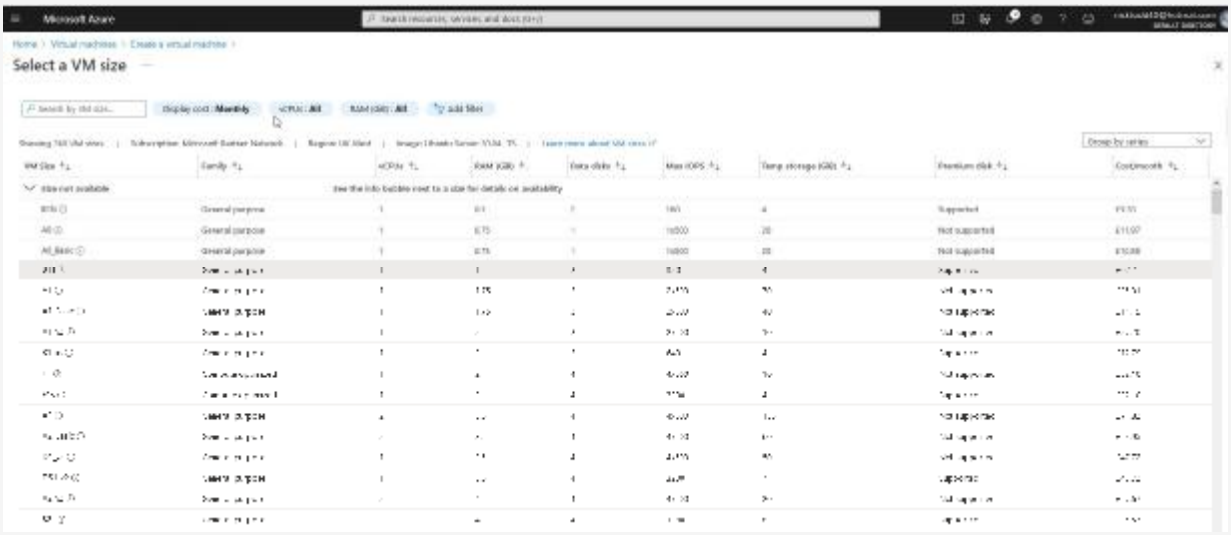
conygre training

Instance Sizing

- Cloud providers all sorts of instance types optimised for things like
  - Memory
  - Compute
  - Graphics
  - Disk Access
  - Graphic cards (for things like Machine Learning)
- You can pick whichever size looks about right
  - And don't forget you can always change the size later if it is too big or too small

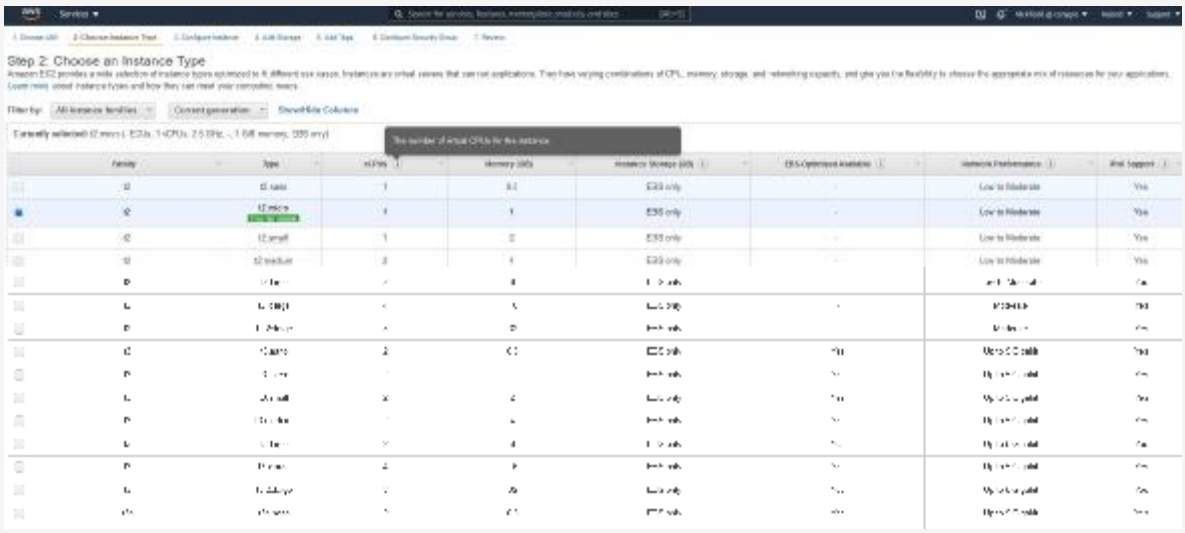
conygre training

Instance Sizing - Azure



conygre training

Instance Sizing - AWS



conygre training

Server LifeCycle

- Servers can be
  - Stopped
  - Restarted
  - Terminated
- Servers and Disks can be resized

conygre training

Networking

- Virtual machines run within a
  - Virtual Private Cloud (VPC) - AWS
  - Virtual Network (VNet) - Azure
- In either case, these are a virtual version of a physical network
- We will introduce the AWS network
  - The other cloud providers operate networks in a similar way

</conygre> training

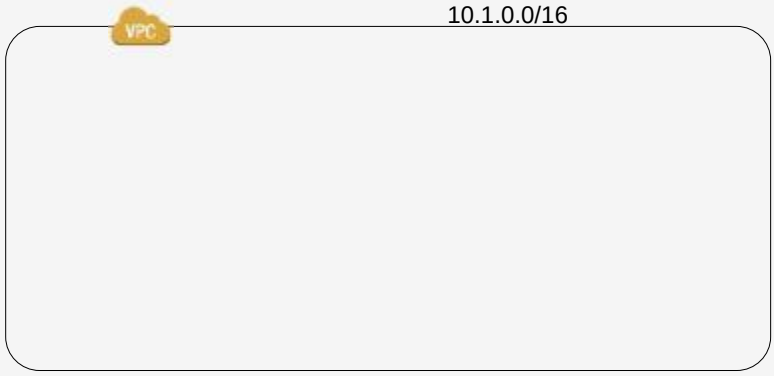
VPC

- The VPC network consists of
  - IP range
  - Subnets within that range
  - Internet Gateways, and NAT servers or gateways
  - VPN Gateways
- A VPC network *spans availability zones* within a region so you can easily run your applications across multiple facilities

</conygre> training

VPC IP Range

- A VPC has an IP address range



CIDR Notation

- VPC network IP addresses are specified using the Classless Inter-Domain Routing (CIDR) notation
- You do need to understand how this works if you are to make sense of VPC configurations

IP Address Basics

- Any IP address is actually a set of four octets (four sets of zeros and ones!)
  - 32 bits in total

`00001010.10001011.10101011.00001011`

- The binary is unmanageable so they are written for us as decimal numbers (0-255)

`10.139.171.11`

</conygre> training

Specifying a Range

- Ranges of addresses is where CIDR comes in
  - Say we wanted all our addresses to start with 10.
- We want to specify that the first 8 bits are fixed, but the remaining 24 bits could be anything

`10.blah.blah.blah`

  - The /8 at the end of this address means that *the first 8 bits are fixed*

`10.0.0.0/8`

</conygre> training

CIDR Examples

- Here is the CIDR for *any* address
  - Zero bits are fixed — 0.0.0.0/0
- Here is the CIDR for 10.2.blah.blah
  - 16 bits are fixed — 10.2.0.0/16
- Here is the CIDR for a specific address
  - All 32 bits are fixed — 10.2.5.123/32

</conygre> training

Further CIDR

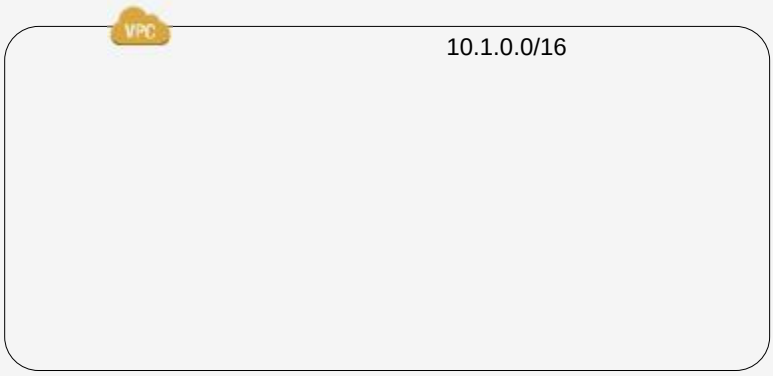
- The specified number of fixed bits does not have to be divisible by 8
  - Makes it a bit more complicated
  - Every increase by one, halves the number of IP addresses available
- For more complex networks you would do that to control address ranges

10.2.0.0/18

</conygre> training

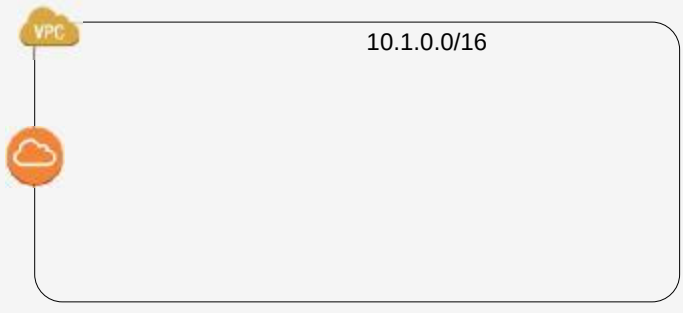
Back to VPC

- In this example, all servers launched into this network will have addresses that begin 10.1



Internet Access

- For any servers in the VPC to have any kind of Internet access at all, the VPC requires an Internet Gateway



Internet Gateway

- The Internet gateway is a free component that your VPC must have to facilitate
  - Internet coming in
  - Internet going out
- For applications that must remain completely off the Internet, you would not use an Internet Gateway

Further Control

- Just creating a VPC with an IP range is not sufficient
- If you launch a machine into the network
  - Which Availability Zone is it in?
  - Is it publicly available on the Internet or not?
  - Can it see other machines in other parts of the network?
- Further control is achieved by subdividing the VPC into subnets

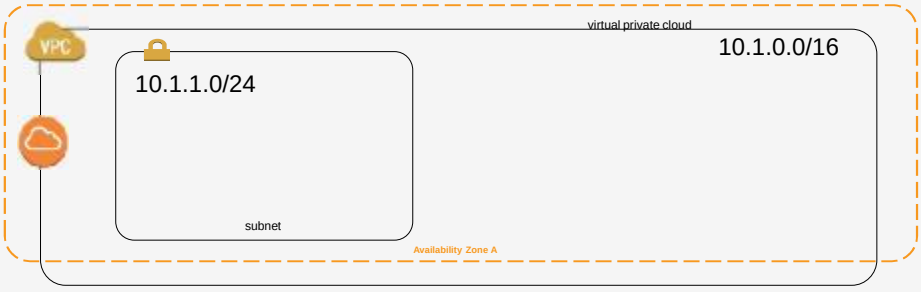
Subnets

- A VPC is broken down into Subnets
- A Subnet is a subset of the IP addresses in the VPC
- A Subnet is
  - Specific to an availability zone
  - On or off the public Internet
    - Controlled through routing rules

conygre training

Subnet Example

- When a subnet is created, you must specify
  - Which availability zone it will go in
  - What IP range it will have
    - This must be a subset of the VPC range



conygre training

Routing

- Routing is used to determine
  - Whether a subnet is on the Internet or not
    - Public or Private in AWS terminology
  - Access over a VPN
  - Access to/from a corporate data centre
  - Access to/from another VPC
- Routing is configured in Routing Tables

Routing Tables

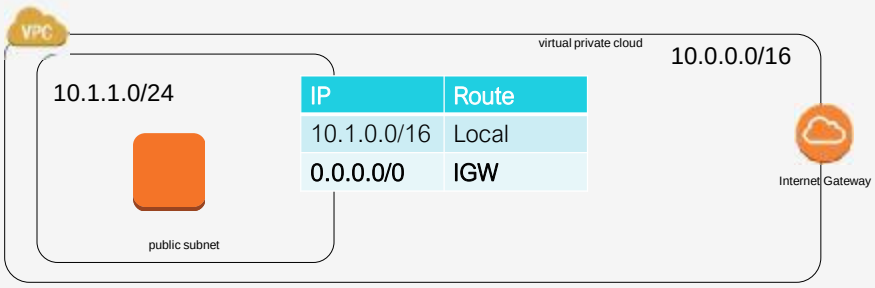
- VPCs have one default routing table entry

CIDR	Route
10.1.0.0/16	local

- This entry means that to access machines with IP addresses matching the VPC IP range, the routing will be local to the VPC

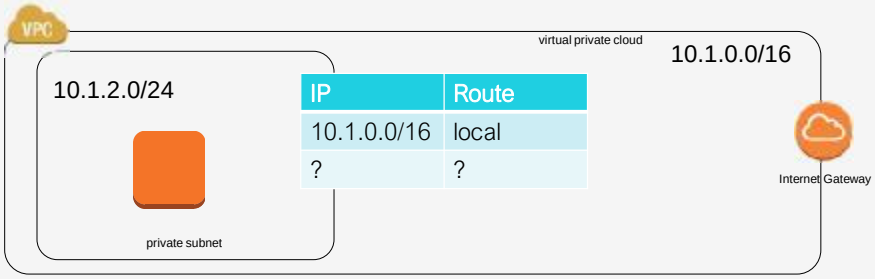
Public Routing

- To allow machines to be accessible *from/to* the Internet, an Internet Gateway is now added to the routing table
  - This subnet is now said to be a *public subnet*



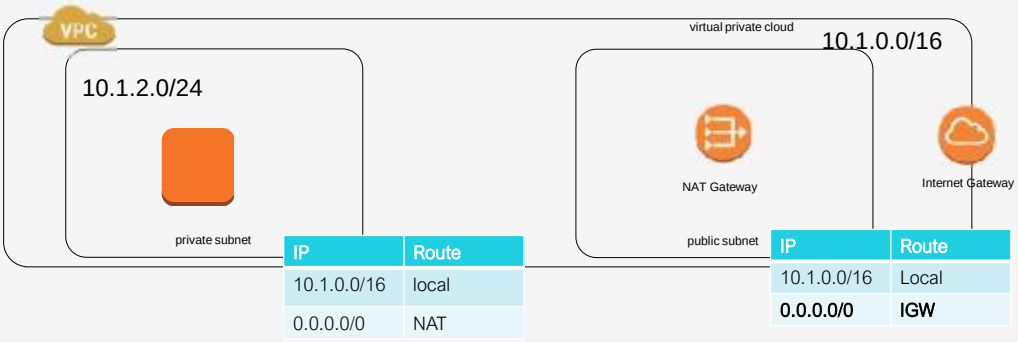
Private Routing

- What about machines that should not be accessible from the Internet?
  - We do NOT use an Internet Gateway
  - However, as it stands, these machines now *cannot access* the Internet!



NAT Gateways

- In order for private machines to access the Internet, a NAT Gateway is used
  - A NAT Gateway allows outbound traffic and the responses, but does NOT allow inbound traffic



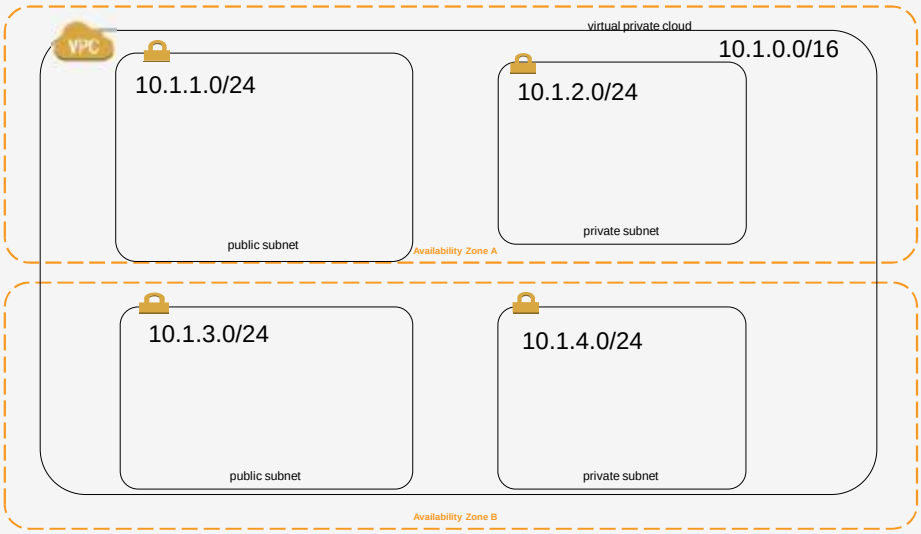
IP	Route
10.1.0.0/16	local
0.0.0.0/0	NAT

IP	Route
10.1.0.0/16	Local
0.0.0.0/0	IGW

High Availability

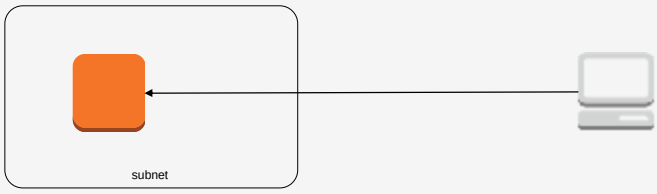
- You now have
  - A public subnet for Internet facing machines
  - A private subnet for private machines
- For high availability you will want replicated machines in more than one availability zone
  - How do we achieve that?
- Create public and private subnets in each zone you want to use

Complete Subnet Example



Controlling Access

- If you launch an EC2 instance in a public subnet with a running Web server on it, will you be able to access the Web site?
  - The short answer is no!



Security Groups – virtual firewalls

- In aviation, firewalls are the sheet of metal between you and the engine!
- The technical term firewall is based on the same concept
- By default all ports on the EC2 are firewalled off
- To open a port such as port 80 for Web you will require a security group

</conygre> training

Security Group Configuration

- Security groups open ports
  - To specific IP ranges using CIDR
  - To other sets of machines that themselves have a specific security group assigned
- The example below opens port 80 and 443 to anywhere

Security group

Type	Protocol	Port Range	Source
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0

</conygre> training

Security Group Examples

- Remote Desktop from Anywhere

Type	Protocol	Port Range	Source
RDP	TCP	3389	0.0.0.0/0

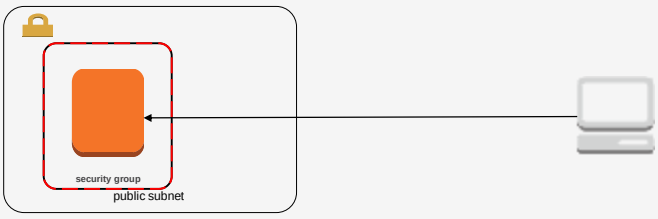
- SSH from a specific address

Type	Protocol	Port Range	Source
SSH	TCP	22	77.101.213.208/32

conygre training

The Web Server

- To make our Web sever available
  - The machine needs to be in a public subnet
  - The machine needs to have a security group assigned with port 80 open



conygre training

Publicly Accessible IPs

- If a machine has an IP address 10.1.2.x, how will that be accessible from the Internet?
  - The machine will also need a publicly accessible IP address
- There are two types of publicly accessible address
  - Public IP
  - Elastic IP

Public IP

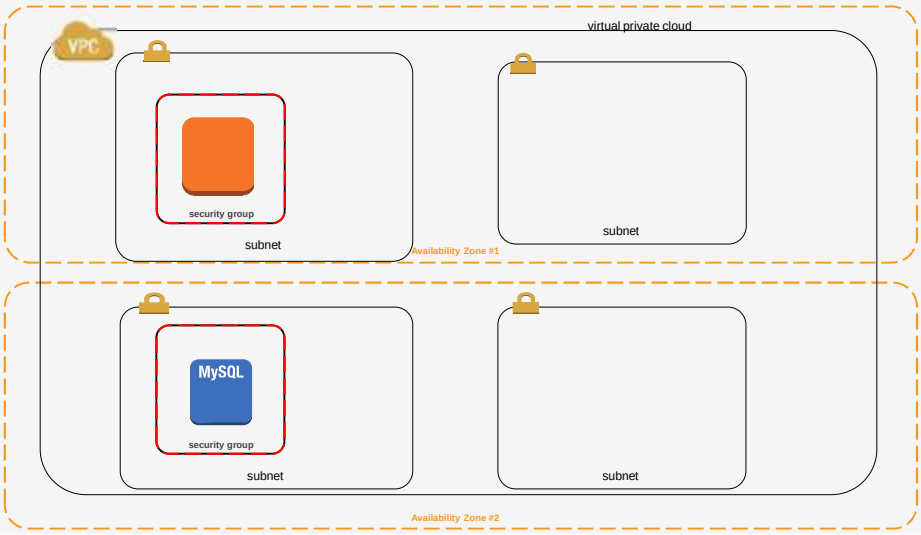
- A public IP can be assigned to an instance
- A public IP address will change if you ever stop and start the instance
  - Could be a problem for your DNS configuration!
- Public IPs are completely free

Elastic IP

- Elastic IP addresses are
  - Fixed
  - Allocated to your account by you
  - Associated with your instances by you
  - Chargeable when they are NOT associated to a running instance
    - Make sure you get rid of any you are not using!

conygre training

VPC Example



conygre training

Additional Features

- VPCs can be connected together using VPC Peering
  - Only if the IP ranges do not overlap
- VPCs can be connected to on premises via
  - VPN Gateway
  - AWS Direct Connect

Summary

- Setting up servers
- Server lifecycle
- Setting up the network

Monitoring and Logging

</conygre training

Objectives

- Monitoring in the Cloud
- Logging in the Cloud
- Alarms and Alerts

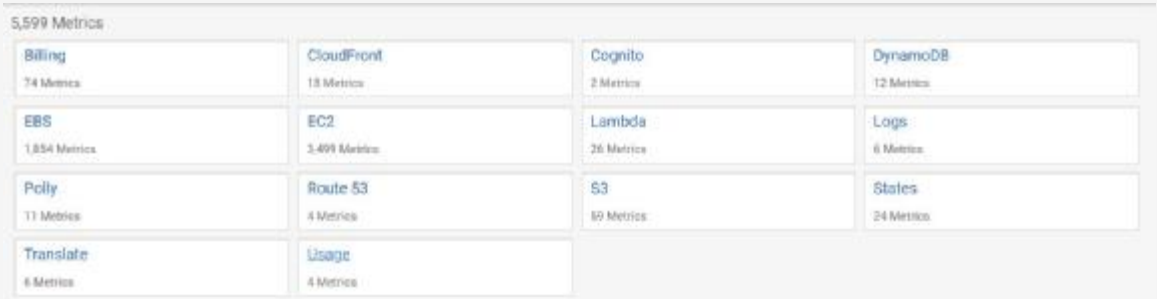
</conygre training

Monitoring in the Cloud

- The core monitoring services
  - CloudWatch on AWS
  - Azure Monitor on Azure
- These services
  - Monitoring data
  - Log files
  - Provides graphical and tabular views of the data
  - Facilitate exportation into third party monitoring products such as an ELK stack or Splunk

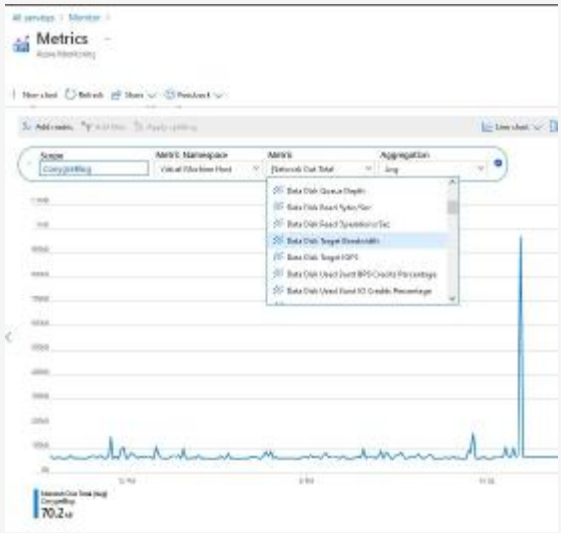
Metrics - AWS

- Metrics are available against all the different services



Metrics - Azure

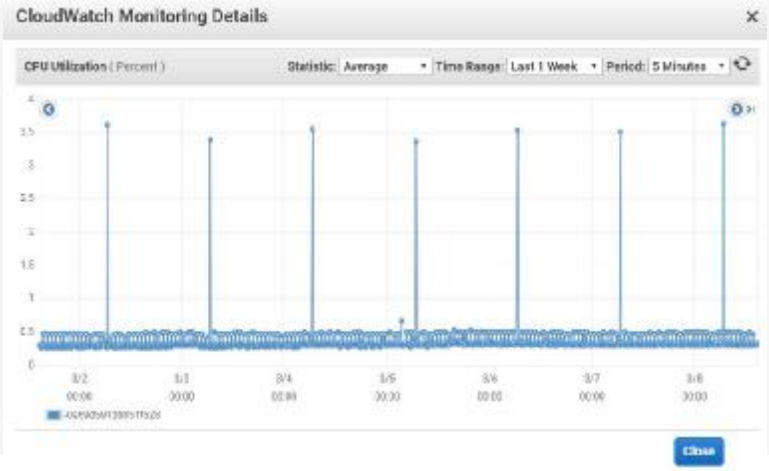
- Here is the Network traffic metric for a virtual machine running a WordPress blog on Azure
- Note that there is a significant number of additional metrics available



conygre training

Drilling Down

- Any metric can be drilled into more detail



conygre training

Virtual Machine Metrics

- On AWS metrics are taken from EC2 instances every 5 minutes
  - You can enable Enhanced Monitoring (at a charge) for 1 minute interval metrics
- On Azure, virtual machine metrics are every minute by default

Custom Metrics

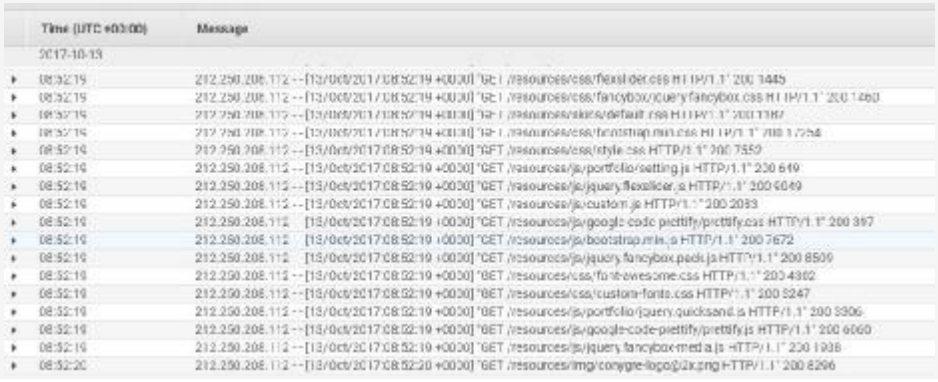
- Applications running on virtual machines can emit any metric value you wish and it will then appear in the monitoring dashboards

Logging

- CloudWatch and Azure Monitor also collate and provide access to logs
- You can also set your own logs to be exported from your applications running on virtual machines
- Logs can be
  - Searched
  - Queried
  - Archived
  - Exported to data stores such as Elastic Search

Custom Logs

- You can install and configure a logging agent on your virtual machines and logs can be exported into CloudWatch or Azure Monitor
- Below is a log output from an EC2 on AWS showing Apache access logs

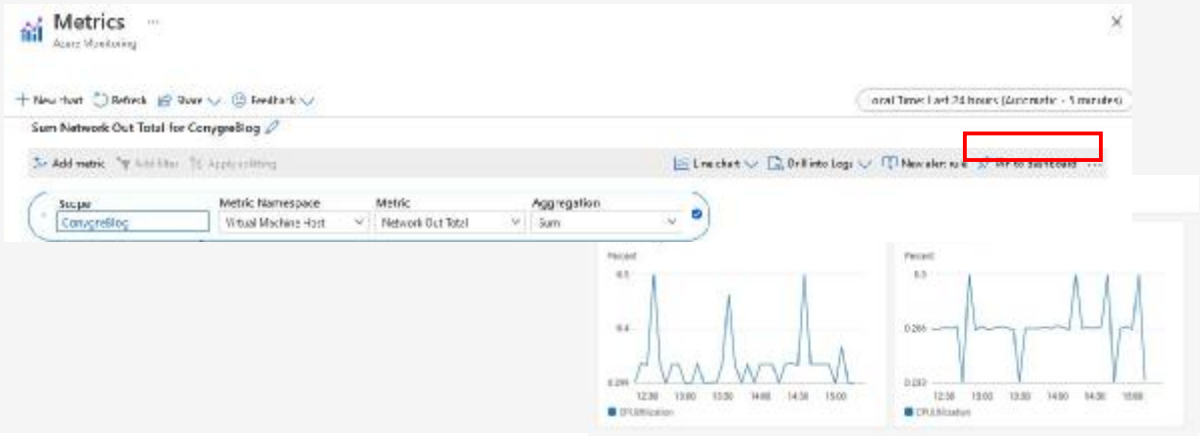


Alarms

- Alarms can be set against metrics or logs
- Alarms can then trigger
  - Notifications
  - Autoscaling events
  - Custom functions to run

Dashboards

- If desired, you can create custom dashboards for your most important metrics and alarms



Exporting Data

- Data can easily be exported into alternative monitoring and logging tools if desired
  - ElasticSearch / Logstash / Kibana (ELK)
  - Splunk

Summary

- Monitoring in the Cloud
- Logging in the Cloud
- Alarms and Alerts

DevOps, Deployment and Automation

</conygre training

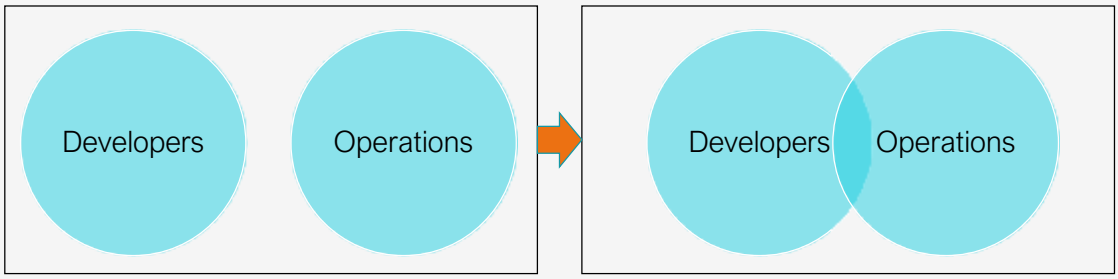
Objectives

- What is DevOps?
- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Infrastructure as Code with Terraform

</conygre training

What is DevOps

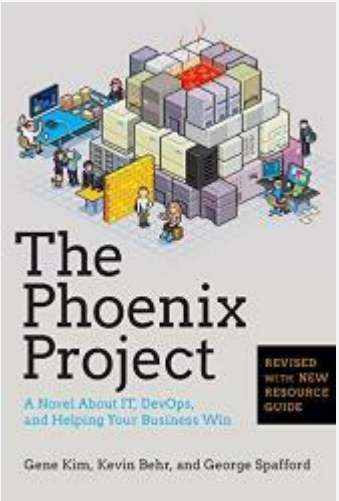
- At its core, DevOps is quite simple
  - Bring developers and operations closer together to take responsibility for building, deploying and monitoring applications



DevOps Benefits

- Shared Responsibility
- Accountability
- More Frequent Deployments
- Better Communication
- Faster to Market

- Read “The Pheonix Project”



Quick Poll

- How often do you deploy into production?
- At least
  - Once a year
  - Every 3 months
  - Every Month
  - Every 2 weeks
  - Every week
  - Multiple times a week
  - Every day
  - Multiple times a day

conygre training

Frequent Deployments

conygre training

Stack Overflow Example

stackoverflow.com

Devops Core Practices

- Continuous Integration
- Continuous Delivery
- Continuous Deployment

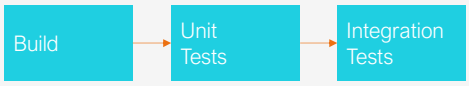
Continuous Integration?

"Continuous Integration is a software development practice where members of a team integrate their work frequently, usually each person integrates at least daily – leading to multiple integrations per day. Each integration is verified by an automated build (including test) to detect integration errors as quickly as possible."

– Martin Fowler

What is Continuous Integration

Continuous Integration



Continuous Integration

- You can use your favorite CI server to continually build and test your code when checked in
- Cloud providers also offer services
  - Azure Devops Pipelines
  - AWS CodeBuild, CodeDeploy and CodePipeline

conygre training

AWS Code Pipelines

- You can use the following services

Service	Purpose
Code Commit	A Git Repository
Code Build	Builds your application
Code Deploy	Deploys your application
Code Pipeline	Create a pipeline of the above
CodeStar	Build the entire pipeline consisting of the above

AWS CodeCommit

AWS CodeBuild

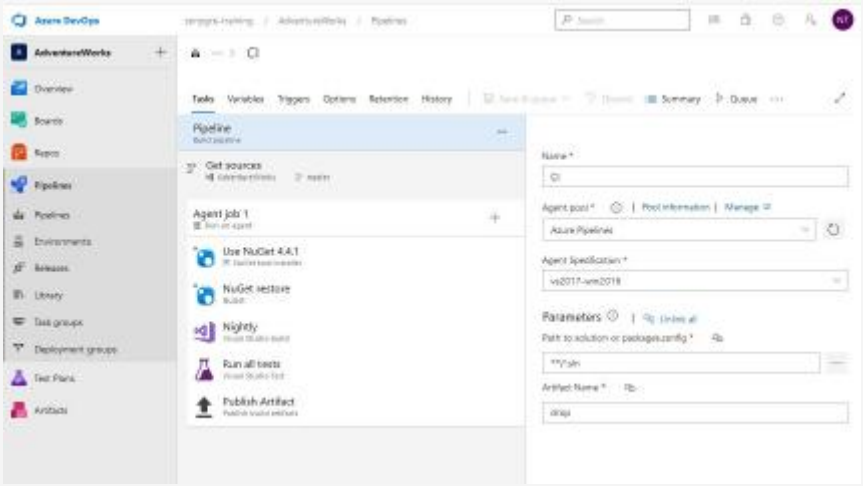
AWS CodeDeploy

conygre training

AWS CodePipeline Example

- Here is a sample pipeline that will build and deploy a machine learning model
  - Source code is in GitHub
  - Build is done by CodeBuild
  - Deploy is done by CodeDeploy
  - The pipeline is created using CodePipeline

Azure Devops Pipelines



What is Continuous Delivery

"The essence of my philosophy to software delivery is to build software so that it is always in a state where it could be put into production. We call this Continuous Delivery because we are continuously running a deployment pipeline that tests if this software is in a state to be delivered."

– Jez Humble, Thoughtworks

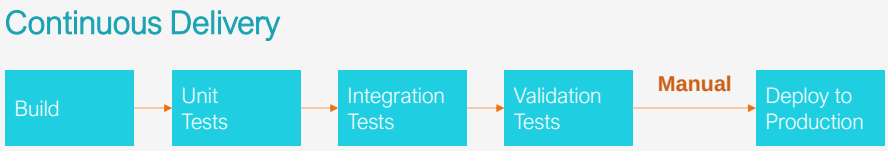
</conygre> training

Continuous Delivery vs Continuous Integration

- Continuous Delivery = Continuous Integration + Fully automated test suite
- Not every change is a release
  - Manual trigger
  - Trigger on a key file (version)
  - Tag releases!
- Continuous Delivery is all about testing!

</conygre> training

What is Continuous Delivery



conygre training

Continuous Delivery

- What changes in the cloud are two key things
  - Where do your deliveries end up?
  - What makes up a delivery?

conygre training

Where Deliveries End Up

- You can place artefacts into object storage ready to be retrieved by your deployment pipeline
- Docker images can be submitted to a docker registry
- Pipelines can also involve the creation of virtual machine images

Deployment

- Deployments to the cloud enable all sorts of capabilities you would otherwise not have
  - Deployment can include the complete creation of a virtual environment
  - Servers don't need to have applications redeployed
    - You simple create new servers for every deployment
- Infrastructure can be in source control as a script
- You can easily create test environments

What is Continuous Deployment

Continuous Delivery

Continuous Deployment

</conygre> training

Continuous Deployment

- When deploying to the cloud, we really want automation
- Frequent, consistent, reliable deployments

- One of the most important ways to achieve this in the cloud is through infrastructure as code

</conygre> training

Infrastructure as Code

- When deploying to a cloud platform or a Docker based environment, you are not creating any physical infrastructure but rather creating virtual environments
- Virtual environments can be created using Web consoles and CLI commands
  - But this is not scripted, and is a manual process that is error prone and slow
- Infrastructure as code is where we create a script that specifies the infrastructure that we want
- Every time we run the script, we get the same infrastructure result
  - Repeatable
  - Consistent
- Since it is a script, it can also go in source control
  - Version controlled infrastructure!!

conygre training

Infrastructure as Code Technology

- There are numerous infrastructure as code technology options
  - AWS have CloudFormation and the Serverless Application Model (SAM) and the Cloud Developer Kit (CDK)
  - Azure have Azure Resource Templates
  - Google Cloud have Deployment Manager
  - In addition for serverless applications there is the Serverless Framework
- Given that there are so many options, in an enterprise environment where multiple cloud providers and deployment variations are in use, it would be good if your teams could work with one consistent technology – enter Hashicorp Terraform

conygre training

Introducing Terraform

- Terraform is a technology that allows you to create infrastructure as code, and it works with multiple different platforms
  - Docker
  - AWS
  - Azure
  - GCP
- Terraform is available from Hashicorp and you can download it from here
  - https://www.terraform.io/downloads.html

Defining a Terraform Deployment

- Terraform infrastructure is defined using HCL which is a configuration language from Hashicorp
  - It looks quite a lot like JSON
  - You can also use JSON
- Terraform configuration files use the extension .tf

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }
}
```

- The main terraform configuration file for your infrastructure is called main.tf

Terraform Single EC2 Example

- Here is a main.tf file that will configure a single EC2 instance on AWS

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }
}

provider "aws" {
  profile = "default"
  region  = "us-west-2"
}

resource "aws_instance" "example" {
  ami           = "ami-08d70e59c07c61a3a"
  instance_type = "t2.micro"
  tags = {
    Name = "ExampleInstance"
  }
}
```

conygre training

Configuring a Provider

- Remember, Terraform supports multiple deployment environments such as AWS and Azure
- Set the required_providers
  - In this case, we are using the Hashicorp AWS provider
  - You should specify a specific version in case a version increment affects your templates
- Configure the provider
  - Since this is AWS, we must set the region and the AWS CLI profile to use

```
terraform {
  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 3.27"
    }
  }
}

provider "aws" {
  profile = "default"
  region  = "us-west-2"
}
```

conygre training

Creating Resources

- The most important entities in any Terraform template file are the resource objects that you define
- Resources are defined as follows

```
resource “resource_type" “your_choice_of_resource_name" {
  property = “value“
}
```

- An EC2 therefore can be defined as follows

```
resource "aws_instance" "example" {
  ami           = "ami-08d70e59c07c61a3a"
  instance_type = "t2.micro"

  tags = {
    Name = "ExampleInstance"
  }
}
```


Creating the Infrastructure

- In order to run our Terraform template, we can run a series of terraform commands
  - terraform init
  - terraform validate
  - terraform plan
  - terraform apply

Deployment Strategies

- Zero-downtime deployment (and roll-back)
- Blue-green
  - Two environments
  - Install on one. Switch. Switch back on problems
- Canary release
  - Deploy to subset of servers
- Real-time application state monitor!

Summary

- What is DevOps
- Continuous Integration
- Continuous Delivery
- Continuous Deployment
- Infrastructure as Code with Terraform

Azure DevOps

</conygre> training

Contents

- Azure Devops Overview
- Terminology used throughout Azure Devops
- Managing Tasks and Reporting Progress
- Repositories
- Pipelines

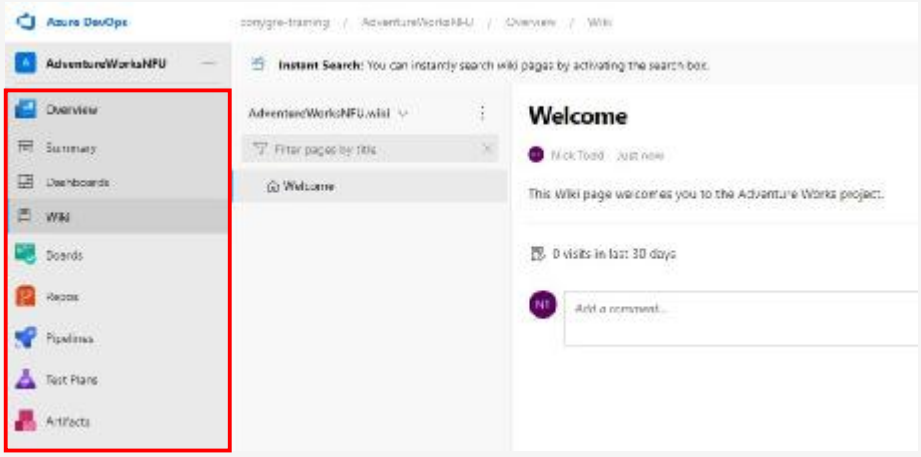
</conygre> training

What is Azure Devops?

- *"Plan smarter, collaborate better and ship faster with a set of modern dev services"*
  - https://azure.microsoft.com/en-gb/services/devops/
- Azure Devops is an evolution of Team Foundation Server from Microsoft that provides a 'one stop shop' for tools that facilitate collaboration and automation around software delivery
  - Azure Devops can be installed locally on your own network or it can be used as a cloud based service
- Microsoft is the only cloud provider with a service like this

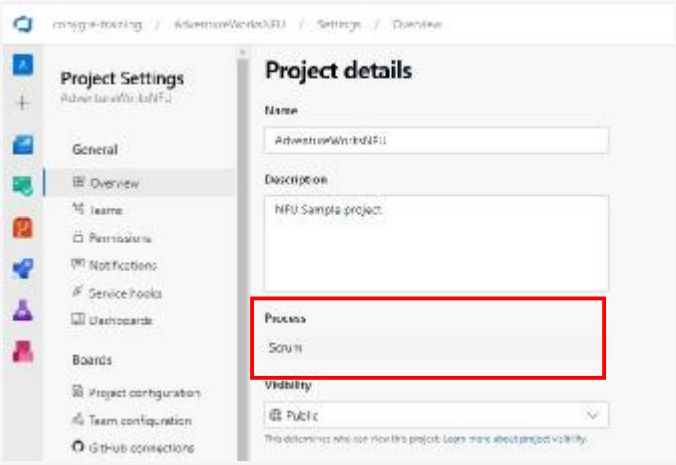
Core AzureDevops Features

- Wiki
- Boards
- Repositories
- Pipelines
- Test plans
- Artifacts



Project Setup

- The Project Setup can be viewed in the Project Settings / Project Details
- The Process can be
  - Scrum
  - Agile
  - CMMI
  - Basic
- The process cannot be changed so choose carefully!
- You can create your own process types by creating a process configuration XML file

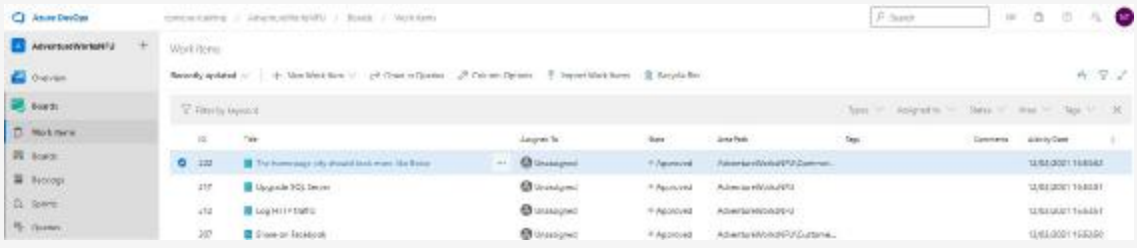


Wiki

- The Wiki area allows you to create informational content surrounding your project
- This can be used for any documentation
  - Risk List
  - Sprint Goals
  - Definition of Done
  - Runbooks
- Content is added using an editor that is provided as part of the interface

Boards

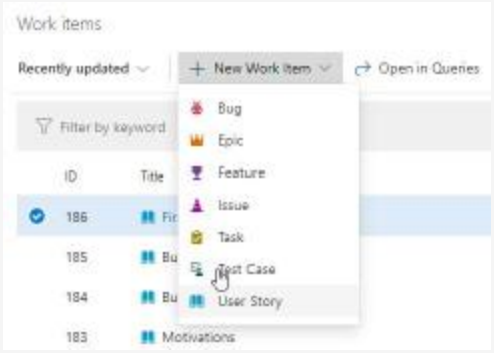
- The **Boards** section is the most important section for project management
- Boards breaks down further into
  - Work items
  - Boards
  - Backlogs
  - Sprints
  - Queries

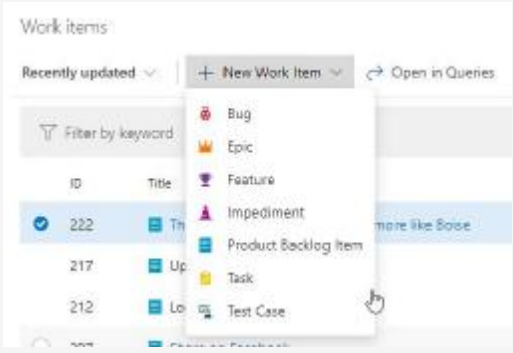


conygre training

Work Items

- Work items are all the items of work than need to be completed
- There can be different kinds of work item, and those types depend on which Process the project was set up with

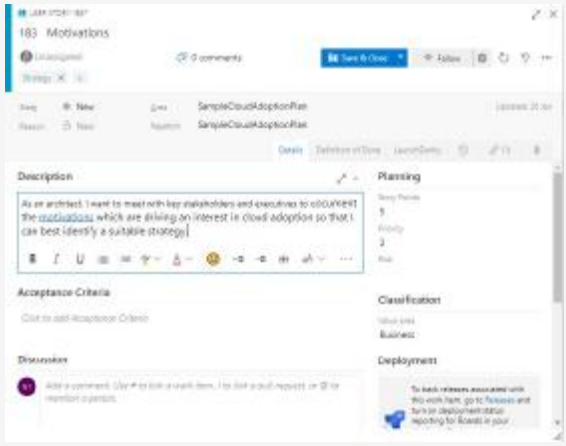




conygre training

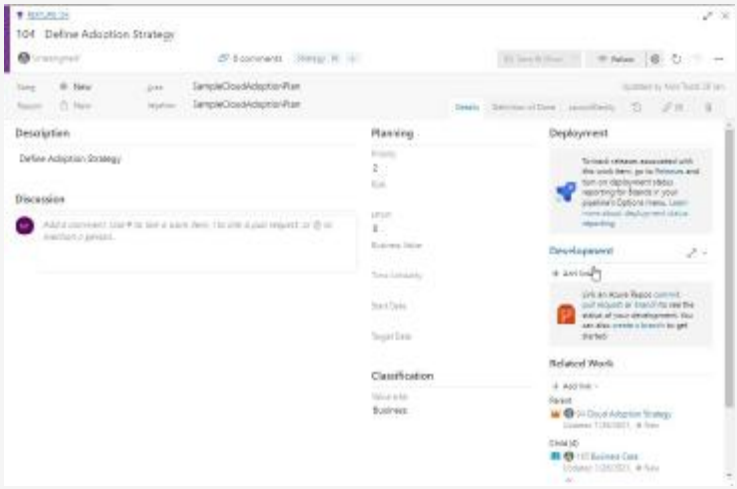
Work Item Attributes - User Story

- Exactly what attributes are present depends upon the template used to create the project but here are some examples
  - Story Points
  - Priority
  - State
  - Area
  - Iteration
  - Description
  - Acceptance Criteria



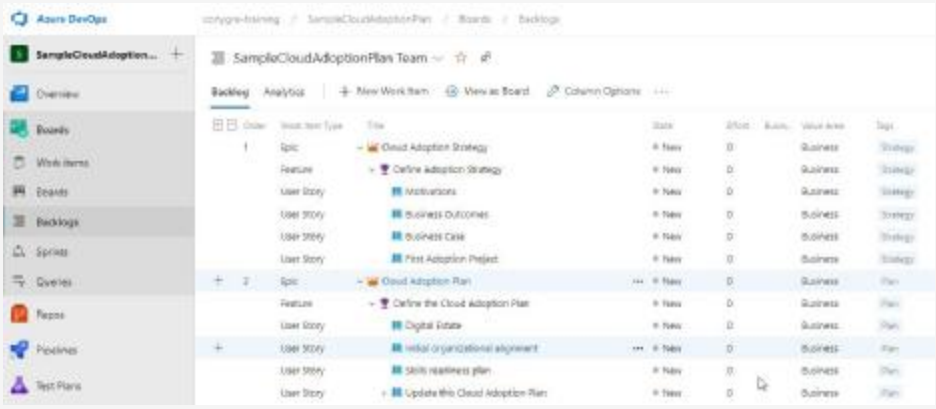
Work Items - Features and Epics

- Features and Epics have attributes such as
  - State
  - Description
  - Priority
  - Risk
  - Effort
  - Business Value



Work Item Relationships

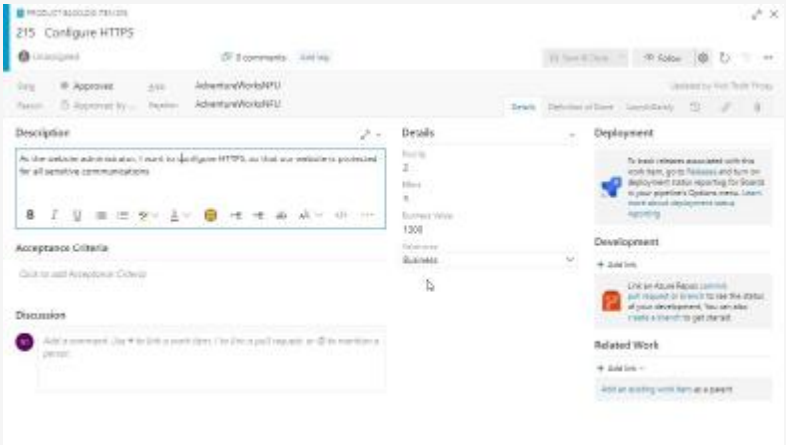
- Work Items can be related to other work items
- Relationships can be viewed in the Backlog view



</conygre> training

Work Items - Product Backlog Items

- Product Backlog Items have
  - State
  - Description
  - Priority
  - Effort
  - Business Value
  - Acceptance Criteria



</conygre> training

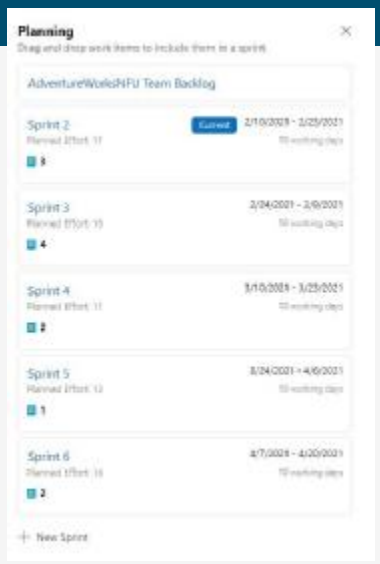
Work Item Relationships

- The exact nature of the relationships between work items depends upon your template, but a typical example would be

- Epics contain
  - Features, which contain
    - User stories which contain
      - Tasks

Sprints

- Stories can easily be allocated to sprints in the Sprints view
- You can drag and drop stories into the preferred sprint
- Each Sprint shows the
  - Planned Dates
  - Number of stories
  - Total allocated story points



Managing Tasks and Reporting Progress

Product Backlog

- List of stories
- Estimated in story points
- More detail on higher priority backlog
- One list for multiple teams
- Product Owner ultimately responsible for priority
- Anyone can contribute
- Maintained and posted visibly
- Derived from Business Plan or Vision Statement, which sometimes have to be created with customer

Sprint Backlog

- Produced as a result of Sprint Planning Session
- Reflects the teams commitment for the sprint
- List of tasks that have been planned for this Sprint
- Prioritised
- Tasks estimated in hours – should be broken down into tasks of less than 16 hours
- Maintained and posted visibly
- Derived from User Stories
- Scrum Team Members responsible for determining tasks in Sprint Backlog

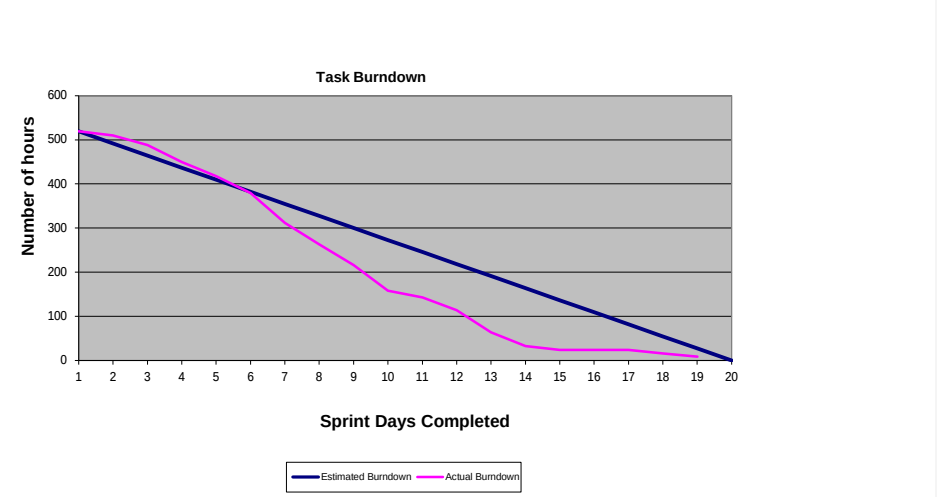
Sprint Backlog

Story	To Do	T… R…			
A user can… 5	Code the … 8 Code the … 5 Test the … 6	X	Code the … SC 6 Code the … DC 4	Code the … LC 2	31
A user can… 2	Code the … 8 Code the … 5				13
A user can… 3	Code the … 3 Code the … 6	X	Code the … MC 4		13

Sprint Burndown

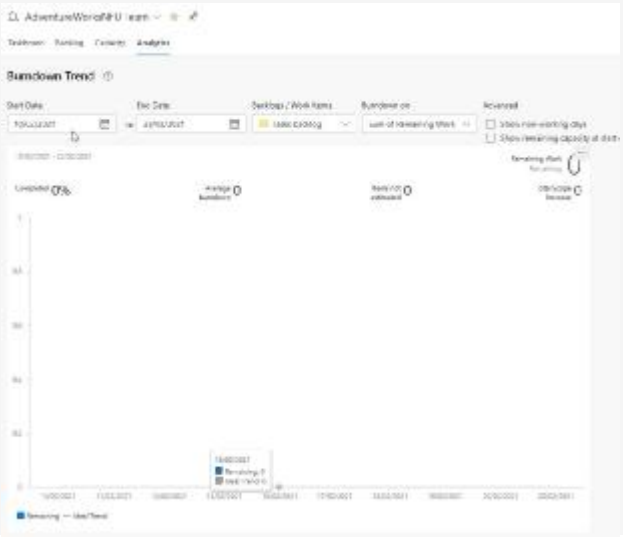
- A tool to guide the development team to successful completion of a Sprint on time with working code that is potentially shippable as a product
- Tracks work remaining against the estimated total
  - NOT effort expended.
- Sprint Backlog updated daily
- Chart can be produced automatically
- Very simple graph of remaining hours estimate v time left
- Big Visible Charts good for feedback, reducing interruptions and showing progress
- Need to be honest!

Sprint Burndown



Burndown in Azure DevOps

- You can find the Burndown under Sprints / Analytics
- IF you keep your task completion up to date in the backlog then this chart will automatically be up to date



conygre training

Kanban Board

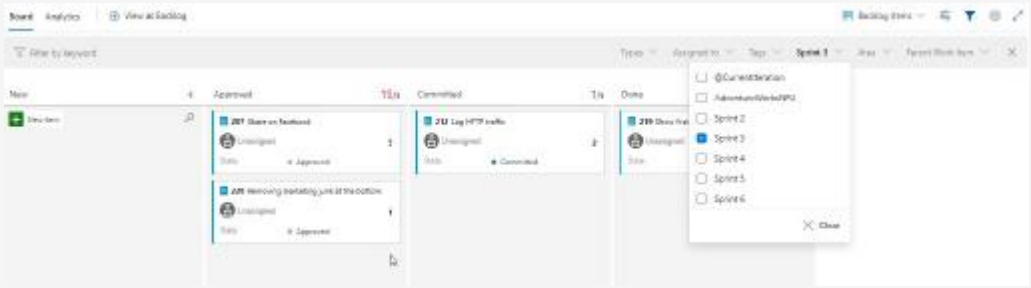
- View progress of getting stories to done
- Identify bottlenecks in your delivery
- You may set work in progress limits on the sections
- What sections are on your board are up to you
  - You can change them as you learn more about how your team works

Basic Kanban Board

Story	Tasks – To do	In Development (6)	In Test (3)	Done
Story A				Task 1
			Task 2	
			Task 3	
		Task 4		
		Task 5		
Story B				Task 6
			Task 7	
		Task 8		
Story C	Task 9			
	Task 10			
	Task 11			
	Task 12			
	Task 13			

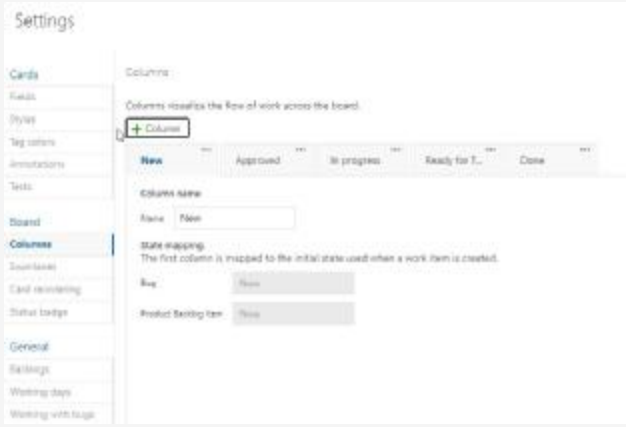
Kanban Boards in Azure Devops

- The Boards section shows the Kanban boards
- The Kanban board for the current sprint will be most relevant



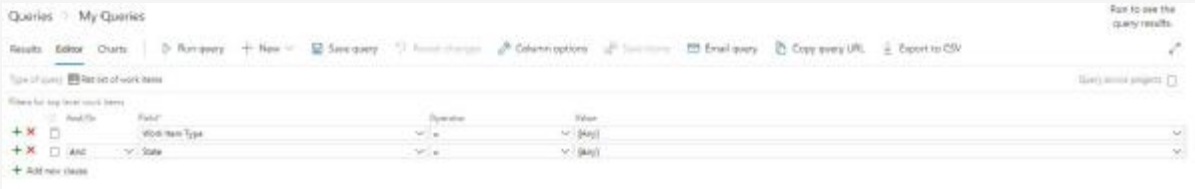
Kanban Board Columns

- Columns can be customised for your project in the Settings



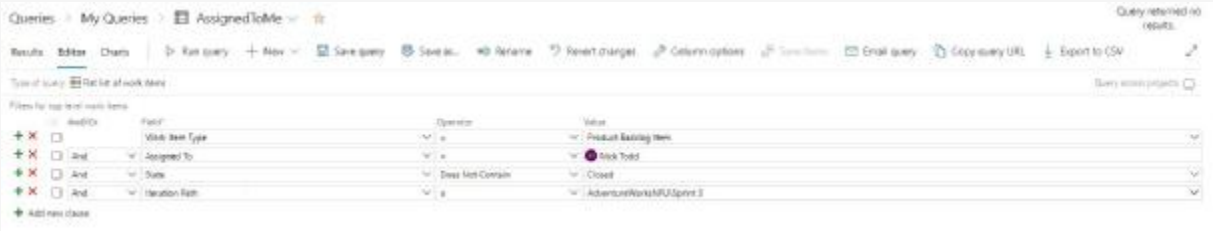
Queries

- The Queries section lets you create queries covering any aspects of your project
- The Queries are created using the Query editor



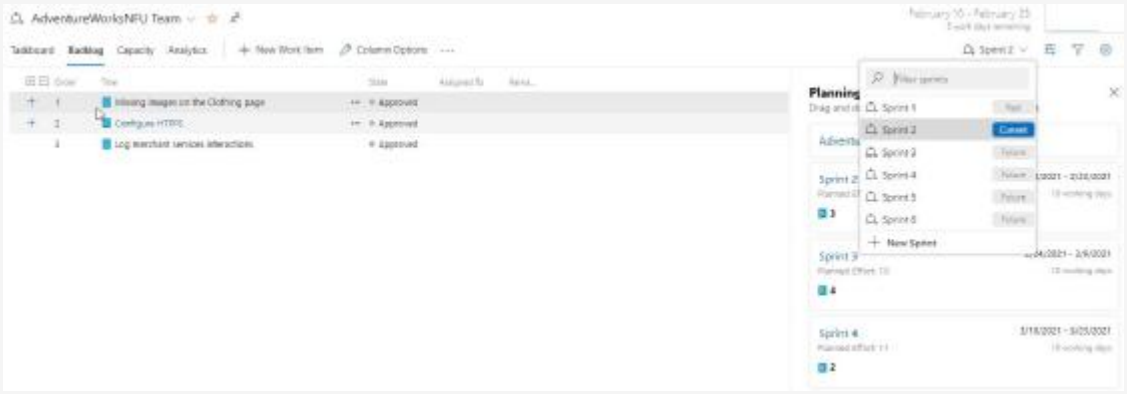
Query Example

- Find all the product backlog items assigned to me for the current Sprint but are not closed



Backlogs

- The Backlog is used to show the backlog of work for the entire release or for the individual sprint

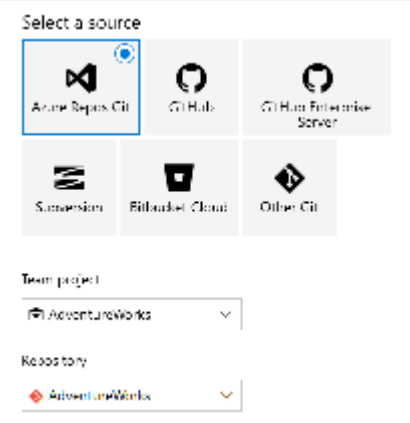


Repos / Repositories

- Repositories are used for source code management by development teams
  - In Azure you can view and manage the code from within Azure Devops
- AWS do have a GIT repository service called CodeCommit
  - AWS also support GitHub and BitBucket

Azure Repository Support

- Azure Pipelines support multiple source code repositories



Pipelines

- Pipelines is where can manage the automation for the build and deploy process for our software
- Pipelines will be discussed elsewhere

Summary

- Azure Devops Overview
- Terminology used throughout Azure Devops
- Managing Tasks and Reporting Progress
- Repositories
- Pipelines

Migrating to the Cloud

</conygre training

Objectives

- Cloud Adoption Strategies
- The Azure Cloud Adoption Strategy
  1. Have a Strategy
  2. Come up with a Plan to implement the Strategy
  3. Ensure that you have set up your technology so you can begin deployments
  4. Begin migrating your applications
  5. Governance
  6. Management

</conygre training

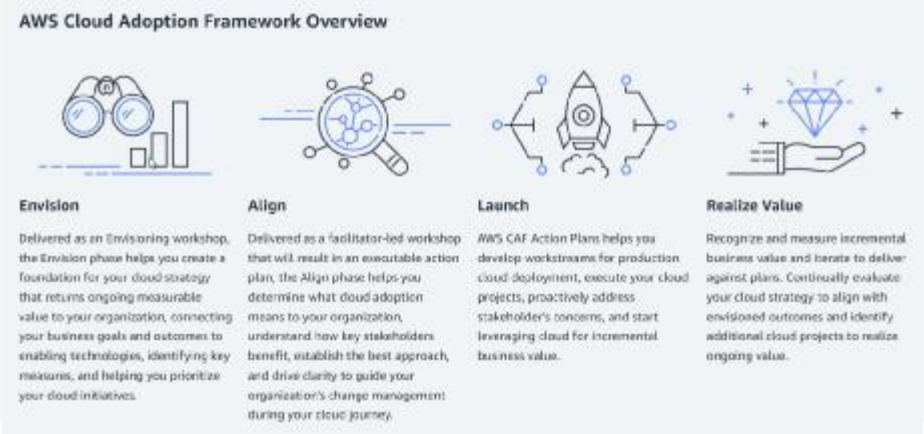
Cloud Adoption Frameworks

- All the public cloud providers have extensive guidance around how to get started and migrate applications and infrastructure to the cloud
- The guidance is generally referred to a Cloud Adoption Framework

- In this module you will be introduced to the Azure Cloud Adoption Framework
  - The others are similar

AWS Cloud Adoption Framework

- The AWS Cloud Adoption Framework can be accessed here
  - https://aws.amazon.com/professional-services/CAF/



The Azure Cloud Adoption Framework

- According to Microsoft
  - *"The Cloud Adoption Framework brings together cloud adoption best practices from Microsoft employees, partners, and customers. It provides a set of tools, guidance, and narratives that help shape technology, business, and people strategies for driving desired business outcomes during your cloud adoption effort."*
- There is a Web site that promotes the best practices and provides explanations to the various components of the framework
  - https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/overview

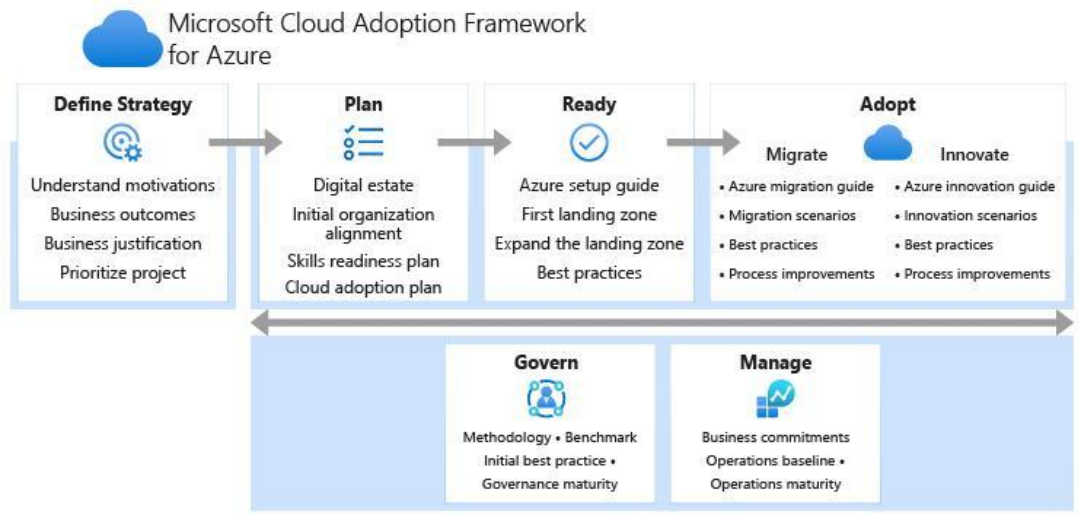
The Framework Components

- The Framework attempts to address a number of issues that come up broadly chronologically
  1. Have a Strategy
  2. Come up with a Plan to implement the Strategy
  3. Ensure that you have set up your technology so you can begin deployments
  4. Begin migrating your applications
- However, they have named these four sections with rather vague words, so it is not obvious!
  1. Define strategy
  2. Plan
  3. Ready
  4. Adopt

Governance and Management

- All the while you are completing the planning, creating the environment, and migrating applications you must also consider your governance and management practices
- Governance and Management are therefore two additional components of the Cloud Adoption strategy

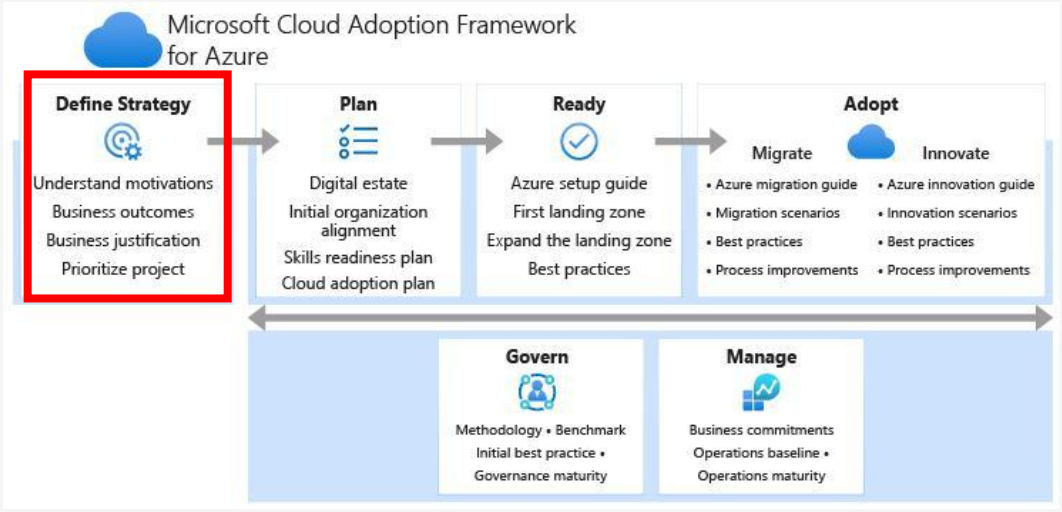
Cloud Adoption Framework from Microsoft



Strategy

The Microsoft Cloud Adoption Framework

</conygre> training



/conygre training

Azure Cloud Adoption Framework - Strategy

- Understanding your Motivations
  - Why are we moving to the cloud?
- Motivations can be around
  - Business events
  - Migration
  - Innovation

Business Motivations

- According to the Microsoft Cloud Adoption Framework, these motivations would include
  - Datacenter exit
  - Merger, acquisition, or divestiture
  - Reduction in capital expenses
  - End of support for mission-critical technologies
  - Response to regulatory compliance changes
  - New data sovereignty requirements
  - Reduction of disruptions and improvement of IT stability
  - Reduce carbon footprint

Migration Motivations

- According to the Microsoft Cloud Adoption Framework, these motivations would include
  - Cost savings
  - Reduction in vendor or technical complexity
  - Optimization of internal operations
  - Increase in business agility
  - Preparation for new technical capabilities
  - Scaling to meet market demands
  - Scaling to meet geographic demands
  - Integration of a complex it portfolio

Innovation Motivations

- According to the Microsoft Cloud Adoption Framework, these motivations would include
  - Preparation for new technical capabilities
  - Building new technical capabilities
  - Scaling to meet market demands
  - Scaling to meet geographic demands
  - Improved customer experiences and engagements
  - Transformation of products or services
  - Market disruption with new products or services
  - Democratization and/or self-service environments

Discussion

- What are your motivations for migrating to the cloud?
- Group them under the headings of
  - Business
  - Migration
  - Innovation

conygre training

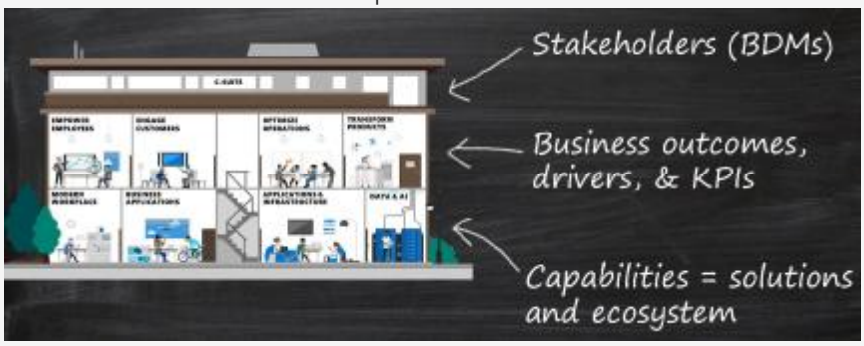
Business Outcomes

- What business outcomes are you looking for
- Any migration to the cloud must consider business outcomes
- You need to consider
  - Fiscal outcomes
  - Agility outcomes
  - Reach outcomes
  - Customer engagement outcomes
  - Performance outcomes
  - Sustainability goals

conygre training

The Business Outcome Template

- Microsoft have provided a template to help you capture your required business outcomes
  - https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/strategy/business-outcomes/business-outcome-template



Discussion

- What business outcomes do you think your organisation is looking for?
- Some areas could be
  - Fiscal outcomes
  - Agility outcomes
  - Reach outcomes
  - Customer engagement outcomes
  - Performance outcomes
  - Sustainability goals

Align Effort to Meaningful Metrics

- Things that can be measured include
  - Number of VMs migrated to the cloud (and related cost savings)
  - Customer behaviour metrics
  - Data innovation metrics

Build the Business Case

- In terms of a business case, you must consider
  - Migration specific initial investment
  - Migration specific revenue deltas
  - Migration specific cost deltas

Choose your First Project

- Your first adoption project should align with your motivations for cloud adoption
- Whenever possible, your first project should also demonstrate progress toward a defined business outcome
- Expectations
  - This project is a source of learning
  - This project might result in production deployments, but it will probably require additional effort first
  - The output of this project is a set of clear requirements to provide a longer-term production solution

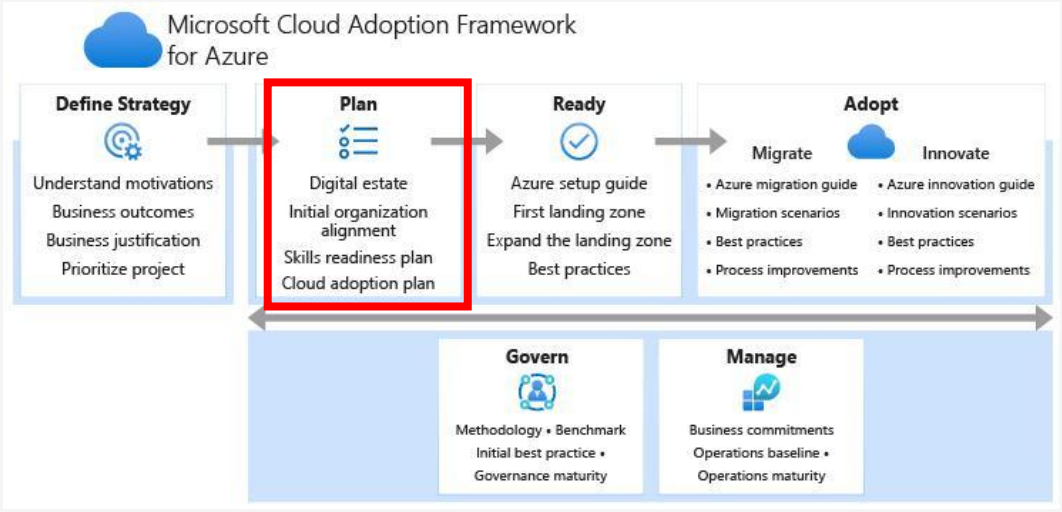
Discussion

- What might be good examples of first projects within your organisation?

Plan

Microsoft Cloud Adoption Framework

conygre training



conygre training

The Planning Process

- The planning process involves the following components
  1. Portfolio Rationalization
     - Create an inventory of your current digital estate and look to see how it could be rationalised
  2. Initial organisation alignment
     - Establish a plan for organisational alignment to support your plan
  3. Skill Readiness Plan
     - Where are the gaps - how will you fill them?
  4. Cloud Adoption Plan
     - Develop a plan to manage change across the portfolio

Portfolio Rationalization

- Microsoft have identified the '5 Rs of rationalisation'
  - Rehost
  - Refactor
  - Rearchitect
  - Rebuild
  - Replace

Organisation Alignment

- You will have to strike a balance between these two sometimes competing concerns
  - Cloud adoption
  - Cloud governance

    - The developer says *"Why can't I just crack on? There are some great services here on Azure!"*
    - The security team says *"No way, we need to have some processes in place to manage this."*
    - The developer then says *"That misses the point of using the cloud!"*

Map People to Capabilities

- The Microsoft Cloud Adoption Framework encourages you to answer the following questions
  - What person (or group of people) will be responsible for completing technical tasks in the cloud adoption plan?
  - What person will be accountable for the team's ability to deliver technical changes?
  - What person (or group of people) will be responsible for implementing protective governance mechanisms?
  - What person will be accountable for the defining those governance controls?
  - Are there other capabilities or people that will have accountability or responsibility within the cloud adoption plan?

Discussion

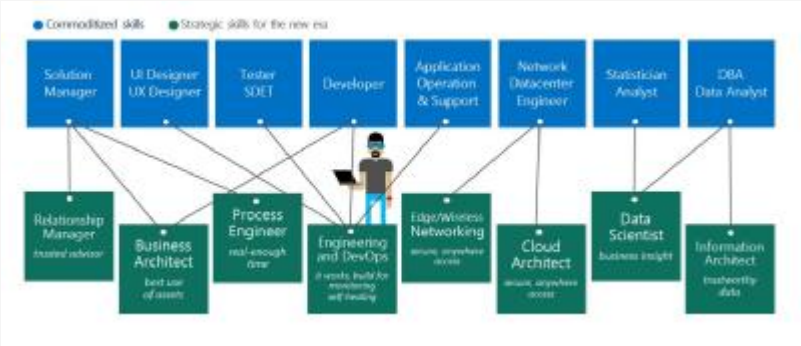
- How can your current organisational structure be adapted for a cloud adoption?
- Can we answer some of these questions now?
  1. What person (or group) will be responsible for completing technical tasks in the cloud adoption plan?
  2. What person will be accountable for the team's ability to deliver technical changes?
  3. What person (or group) will be responsible for implementing protective governance mechanisms?
  4. What person will be accountable for the defining those governance controls?
  5. Are there other capabilities or people that will have accountability or responsibility within the cloud adoption plan?

Build a Readiness Plan

- Staff roles will change
- You will need to
  - Capture and address staff concerns
    - "what is going to happen to my job?"
  - Identify gaps
  - Partner across teams - similar to the move towards DevOps

Mapping Roles to Skills

- This diagram from Microsoft shows relationships between commoditised skills and cloud skills



https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/plan/suggested-skills

Discussion

- What concerns have you picked up within your organisation, and what concerns do you think will come up?
- How mature is your approach to
  - DevOps
  - Agile
- What knowledge gaps currently exist?

Create a Cloud Adoption Plan

- You can now plan out what you plan to do
- Microsoft suggest that must consider Strategic and Tactical inputs

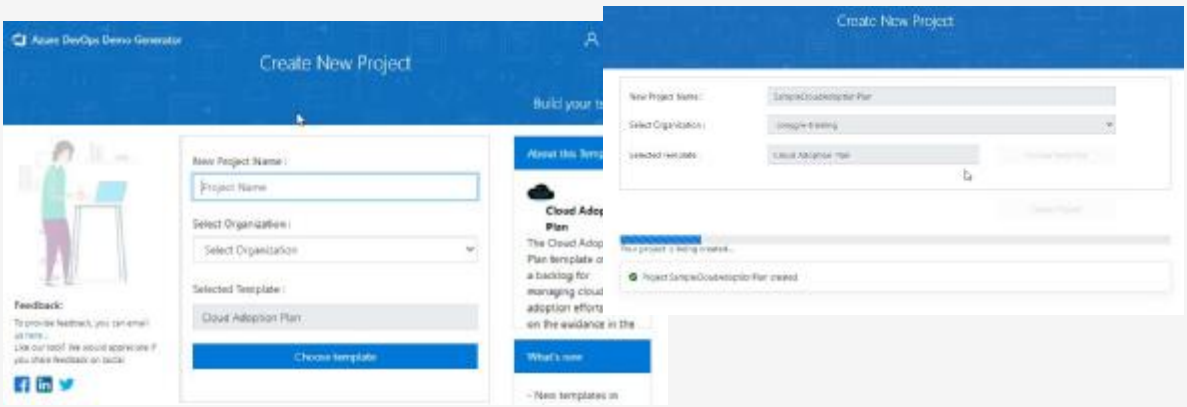
- Strategic Inputs
  - Clear motivations
  - Defined business outcomes
  - Business justification

- Tactical inputs
  - Digital estate rationalisation
  - Organanisational alignment
  - Skills readiness

Planning and Azure Devops

- If you use AzureDevops you can create the plan as an AzureDevops project
- Microsoft have created a suite of templates based on the Microsoft Cloud Adoption Framework to speed up the process
- This can be created using the Azure Devops Demo Generator
  - https://azuredevopsdemogenerator.azurewebsites.net/?name=CloudAdoptionPlan

Azure Devops Demo Generator



Azure Devops Templates

- There are several templates that you can select from that will give you a starting point for your Cloud Adoption Plan

Editing Your Plan

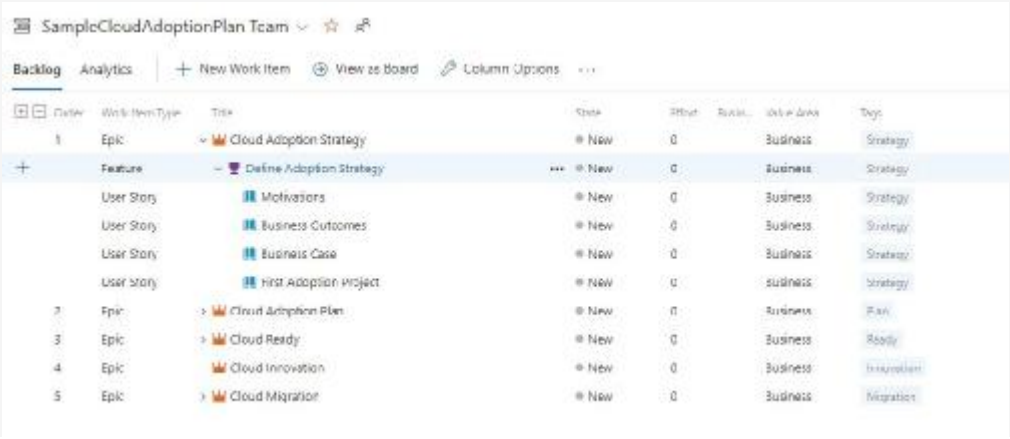
- The plan is created in AzureDevops
- Since it is an AzureDevops project, the plan can now be edited in three main ways
  - Via the AzureDevops Web console
  - Bulk edits via Excel
  - Using Microsoft Project

The AzureDevops Project Plan

- The generated plan uses Agile terminology for the various artifacts
  - Epics
  - Features
  - User Stories
  - Tasks
- The Epics backlog view gives the clearest overview of what needs to be done
- Note that what is generated focuses on migration
- You will need to manually complete the governance requirements

Viewing the Epics, Features and Stories in Context

- In the dropdown on the right, select Epics, and then expand the Epics to see the Features, and then expand the Features to see the User Stories



The Epics in the Template

- There are five Epics in the template
  - Cloud Adoption Strategy
  - Cloud Adoption Plan
  - Cloud Ready
  - Cloud Innovation
  - Cloud Migration

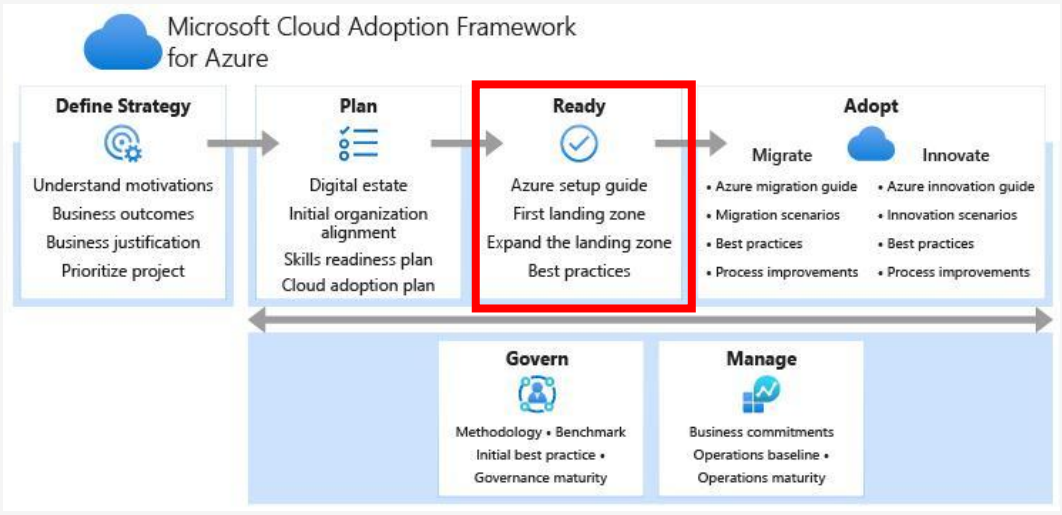
- The Migration epic is where we define the plan regarding migrating actual infrastructure

Adapting the Cloud Migration Epic

- Workloads
  - Features in the cloud migration epic should capture each workload to be migrated
  - Focus on your top 10 workloads
- Assets
  - Each asset (VM, application, or data store) is represented by the user stories under each workload
  - Add and modify those user stories to align with your digital estate
- Rationalization
  - As each workload is defined, you may decide to make changes to those assets
- Create release plans
  - Map releases to your iterations
- Establish timelines

Ready

Create a Landing Zone to Host the Workloads



Azure Landing Zones

- Cloud providers like Azure and AWS have the concept of Landing Zones
- Without landing zones, you could open a cloud account, but then how will you set it up for multi-user access and to host multiple applications at scale?

- Landing Zones, are provisioned with many best practices around networking and security for example already in place

Landing Zone Design Areas

- There are multiple design aspects to consider for your landing zone, and these are referred to as Design Areas
  - Enterprise Enrolment
  - Identity
  - Network topology
  - Resource organisation
  - Governance
  - Operations baseline
  - Business continuity and disaster recovery
  - Deployment options
- In reality these areas will evolve and develop as you learn what works best for your organisation

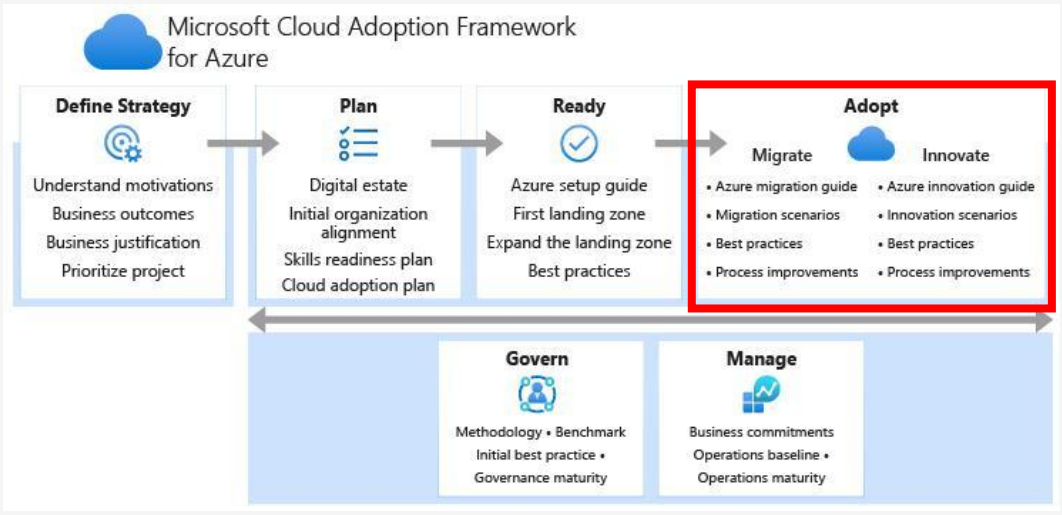
Landing Zone as Code

- One of the paradigm shifts you have to get used to with the cloud is that your infrastructure is now code
- Infrastructure as code, means that it is easy to
  - Destroy it
  - Change it
  - Create copies of it
- This means that landing zone can also be changed, so decisions you make now are not fixed
  - You don't have to get everything right on migration 1, but rather you can evolve your landing zone as you deploy further migrations
- https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/ready/landing-zone/refactor

Infrastructure as Code

- The Azure service for this is called the Azure Resource Manager (ARM)
- Using ARM templates means that your infrastructure can be in source control, potentially along with the application itself!
- Plenty of examples can be seen here
  - https://github.com/Azure/azure-quickstart-templates
- Below is a single VM Wordpress deployment example
  - https://github.com/Azure/azure-quickstart-templates/blob/master/wordpress-single-vm-ubuntu/azuredeploy.json

Adopt



Adopt

- According to Microsoft, Adopt covers migration and innovation
- Migrate
  - Migrate and modernize existing workflows
- Innovate
  - Develop cloud native or hybrid solutions
- In other words, lift and shift type approaches along with refactoring and reengineering approaches

Migration Considerations

- When looking to migrate resources and applications to the cloud, you must consider
  - Technical fit
  - How you will actually perform the migration
  - How you will manage costs
  - How you will optimise for cost and performance once deployed

The Azure Migration Service

- The Azure Migration Service can help achieve the following
  - Assess migration suitability
  - Perform performance sizing
  - Provide cost estimates

Innovate

- Innovation is where you can potentially unlock the greatest business value
- Before you innovate, you will need to
  - Manage customer feedback
  - Democratise data
    - Azure Data Catalog
  - Engage via applications
    - Azure App Service
  - Empower adoption
    - Implement effective DevOps solutions
  - Interact through devices
    - Industrial Assets
    - IoT
  - Predict and influence
    - AI and ML

- https://docs.microsoft.com/en-us/azure/cloud-adoption-framework/innovate/innovation-guide/

Innovation Considerations

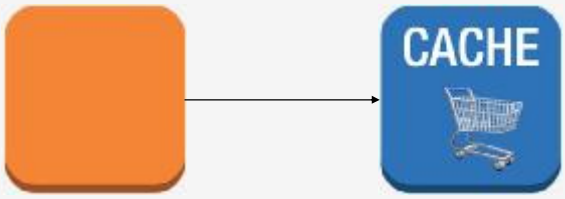
- Cloud based architectures *significantly change* how you approach development
- Key changes include
  - Stateless servers
  - Decoupled infrastructure
  - Horizontal scaling wherever possible
- To illustrate we will use the example of a standard shopping Web site

Session Data

- Many Web sites store session state like shopping carts
  - Where do you keep it?
- If you keep it on the server, this prevents elasticity
  - Servers cannot be destroyed in response to reduced load if people have shopping carts on them

Session Data Options

- A far better place for session data could be
  - CosmosDB
  - Azure Caches for Redis

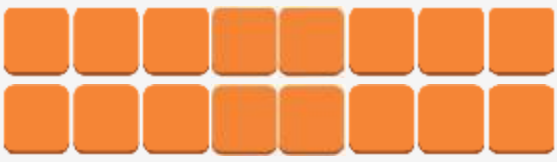


Vertical Scaling

- Vertical scaling is 'make the server bigger'
- Vertical scaling always has limits
  - A machine can't simply keep getting bigger!

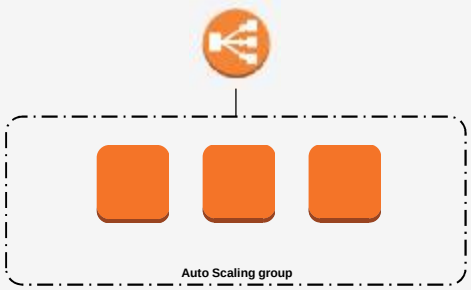
Horizontal Scaling

- Horizontal scaling is 'have more servers'
- Horizontal scaling is far less limiting
  - Consider a scenario where two servers are running at 60%
    - If one fails, then the remaining one is now at 120%
  - If you have 20 small servers
    - If one fails, the others will hardly notice
  - Multiple smaller servers is also generally cheaper than a small number of large servers



Horizontal Scaling on Azure

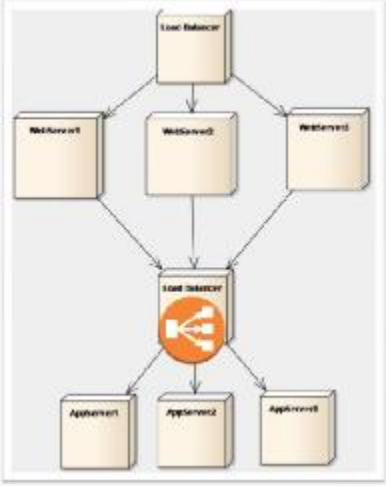
- Virtual Machine Scale sets facilitate horizontal scaling



Number of servers Increases and decreases based on metrics decided by you

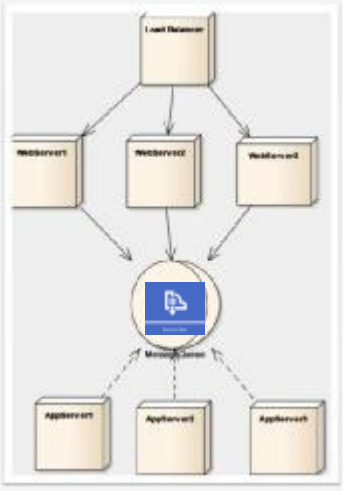
Loose Coupling

- Tightly coupled servers are unnecessarily inflexible
- Direct servers to load balancers NOT server IP addresses



Messaging

- Loose coupling through messaging is also effective
- Azure provide the Azure Service Bus to facilitate messaging



Serverless Computing

- Consider the code that might need to be invoked when someone uses a chatbot
  - You might run a server somewhere hosting an API used by the chatbot
  - You would be paying for that server 24/7 and would need to ensure that it was highly available (so you would run at least two servers)
  - You would also need to ensure that it could scale in response to load
  - New deployments could involve updating these servers every time

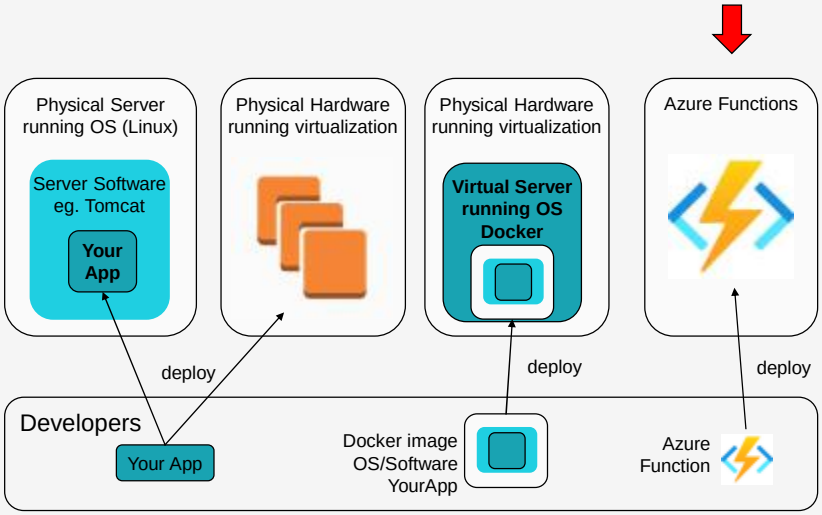
Ditch the Server!

- What if you could write the code that provided the backend for your chatbot and then *deploy the code*
  - The cloud provider could manage the infrastructure that it runs on
  - You could then focus on writing the code
- No servers to
  - Pay for
  - Maintain / Patch
  - Provide a possible hacker entry
  - Scale up and down

Introducing Azure Functions

- Azure Functions allow you to create serverless architectures
- In the serverless paradigm, you would create a function that could be invoked when someone uses your chatbot
- The function is all that you deploy!
  - No servers
  - No maintenance
  - No ongoing server costs
  - No hassle!

Deployment Evolution



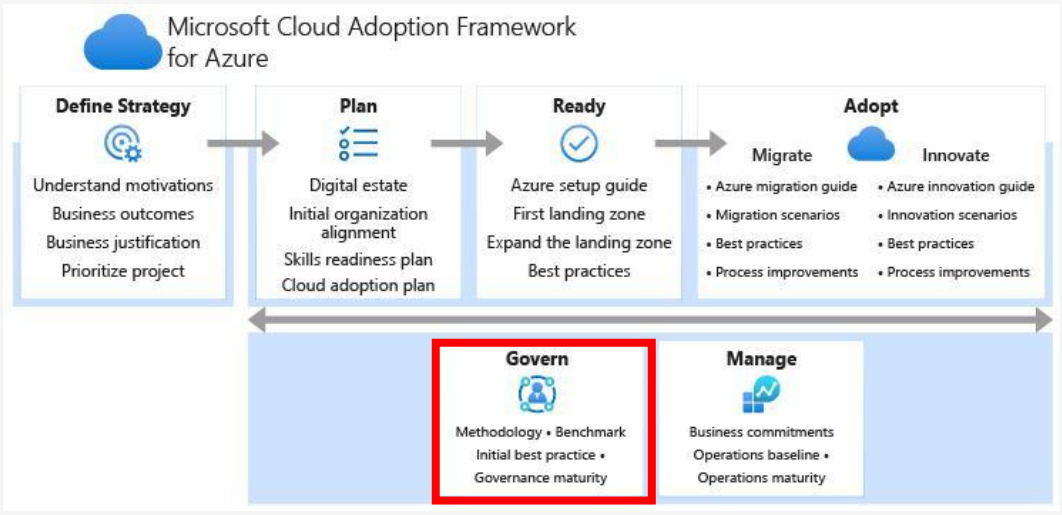
Azure Functions

- Currently on Azure, Functions can be written using
  - Java
  - Python
  - NodeJS
  - .NET Core
  - Powershell Core
  - Custom Runtime

Discussion

- Can you think of examples of applications in your current estate that would lend themselves to
  - A simple 'lift and shift'
  - A Serverless Rearchitecting
  - A Server based rearchitecting involving decoupling facilitating horizontal scaling
  - A Container based solution

Govern



Governance

- Governance in the cloud is different to governance on premises
- Without governance, you will get
  - A proliferation of technology implementations
  - A proliferation of practices and approaches
  - A proliferation of costs!

Governance Exercises

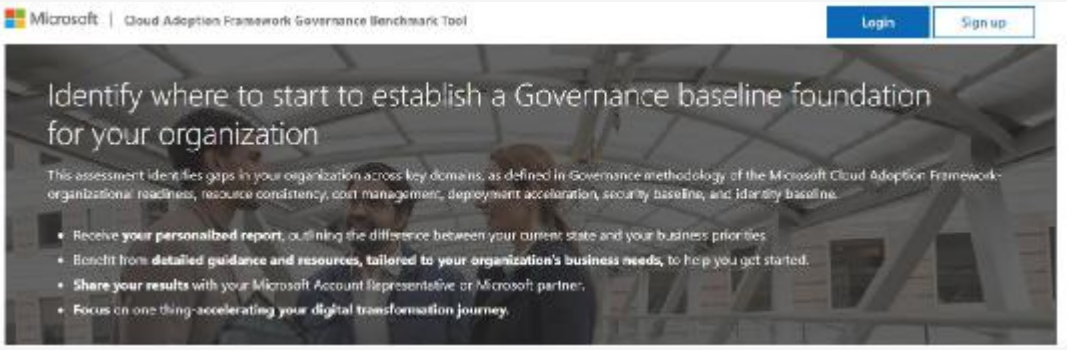
- The Microsoft Cloud Adoption Framework recommends you complete the following exercises
  1. Establish a basic methodology for cloud governance
  2. Benchmark your current and future states
  3. Begin small with a easy to implement set of governance tools
  4. Iteratively improve on your foundation

Governance Methodology

- Governance needs to grow along with your cloud adoption
- It would be impossible to create all of the required guardrails before you get going
- Begin with a minimally acceptable governance policy around things like
  - Identity
  - Security
  - Deployment acceleration
- Iteratively evolve the governance methodology as you move forward with your cloud adoption

Governance Baseline Foundation Tool

- Microsoft have created a tool to help you to baseline your governance foundation
  - cafbaseline.com



Initial Governance Foundation

- To help create your initial governance foundation, you can use one of two governance guides from the CAF portal
  - Standard Governance Guide
  - Governance Guide for Complex Enterprises

Governance Improvement

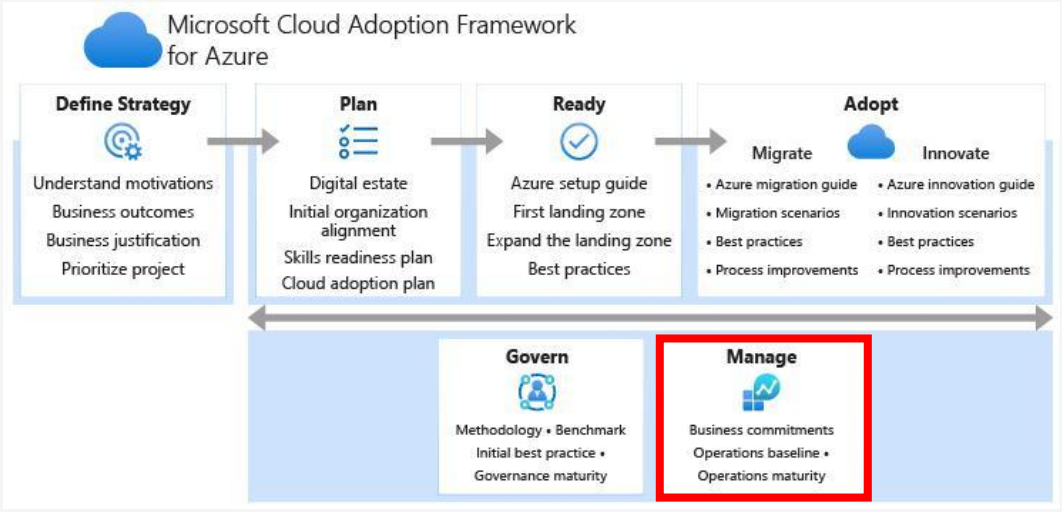
- Governance can become more sophisticated around areas such as
  - Sensitive data in the cloud
  - Mission critical applications in the cloud
  - Cloud cost management
  - Multicloud
  - Complex or legacy identity management
  - Multiple layers of governance

Discussion

- One simple and important decision is around how to tag resources
  - What tags do you think will be important for your resources in the cloud?

Manage

conygre training



conygre training

Managing your Cloud Adoption

- Define Business Commitments
- Establish a Management Baseline
- Expand the management Baseline
- Consider design principles and architectural reviews

Defining Business Commitments

- You will need to ensure that business and IT understand the new terminology
- Business commitments need to be defined in terms that everyone understands
- Areas of these commitments will include
  - Criticality
  - Impact
  - Commitment

Management Baseline

- According to the CAF
  - *"A management baseline is the minimum set of tools and processes that should be applied to every asset in an environment"*
- The cloud provides an opportunity to greatly improve on these aspects of your infrastructure
  - Monitoring of Azure services
  - Log centralisation
  - Monitoring centralisation
  - VM inventory and change tracking
  - Subscription monitoring
  - Guest OS monitoring
  - Network monitoring
  - DNS monitoring
- Azure provide services to help and facilitate all of these

Expand the Management Baseline

- Once you have your basic management baseline in place, you can look to enhance what you have and look to improve where there are business requirements
  - Inventory and visibility
  - Operational compliance
  - Protect and recover
  - Platform operations
    - Databases and Kubernetes services
  - Workload operations
    - Resizing and remediating and protecting

System Design

- System design can become more specialised
- Consider the Cloud Well Architected Framework and the five pillars
  - Cost optimisation
  - Operational excellence
  - Performance efficiency
  - Reliability
  - Security
- https://docs.microsoft.com/en-us/azure/architecture/framework

Discussion

- What do you currently use to manage and monitor your resources?
- How do you think it might be best to integrate with the solutions provided by the cloud?

Summary

- Cloud Adoption Strategies
- The Azure Cloud Adoption Strategy
  1. Have a Strategy
  2. Come up with a Plan to implement the Strategy
  3. Ensure that you have set up your technology so you can begin deployments
  4. Begin migrating your applications
  5. Governance
  6. Management

Managing Costs

</conygre training

Objectives

- Is the Cloud Cheaper?
- Good Cloud Governance
- Good Cloud Accountability
- Cloud Provider Utilities for Cost Management

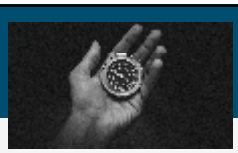
</conygre training

Is the Cloud Cheaper?

- The simple answer is 'usually yes'
- You only pay for what you use
  - If you don't need something to be on you can turn it off or terminate it
  - You can resize things that are bigger than are needed
  - You can take advantage of autoscaling
  - Some cloud services make certain capabilities much cheaper
    - Serverless compute for example

conygre training

Cloud Governance



- The key to ensuring that it ends up being cheaper for you, is to ensure that you have good cloud governance
- Here are some examples of what can go wrong
  - Easy to Create Stuff
    - It's so easy to create stuff in the cloud, your team members keep creating new stuff and forget / fail to remove it
  - 24/7
    - Stuff is left switched on even when not in use
  - What's What?
    - Resources are not clearly identified, so you don't know what you can safely delete
  - Accountability
    - There is no clear accountability around billing, so there is no culture of keeping costs down

conygre training

Easy to Create Stuff

- The Cloud Platforms all make it easy for you to create resources
- You need to build in some governance around how this is done
- This is a difficult balance
  - Making it really easy can result in significant cost quite rapidly
  - Putting in too many controls will result in frustration and will slow down innovation
- Mitigations
  - Create a central team who help manage best practices for other teams to adopt
  - Create guard rails in your accounts that prevent any excesses
  - Create rules such as 'nothing can be created without a cost centre tag on it'

</conygre> training

24/7

- One of the benefits of the cloud is that you can turn stuff off that you are not using
- Make sure you actually do though!
- Some cloud users still think like it is a data centre where everything is on all the time
  - This will incur significant costs
- Consider an application used 9-5 Mon-Fri - that is 40 hours
  - If you leave it on 24/7 that is 168 hours!
  - The cost difference is more than 4x !!
- To enforce this, you can create stopinators which are simple scripts that turn stuff off
  - Doing this alone can make an enormous difference to your cloud costs

</conygre> training

What's What?

- Over time you could end up with a significant number of resources being created in your account
  - But what is everything?
- There are several strategies that can be adopted
  - Multiple cloud accounts
  - Tag resources

Multiple Cloud Accounts

- When working with multiple accounts, this can make it much easier to track resources
- Different departments have their own accounts making it easy to track costs
- Some organisations have different accounts for
  - Dev
  - Test
  - Prod
- However you slice it up, it can significantly help with cost management

Tagging

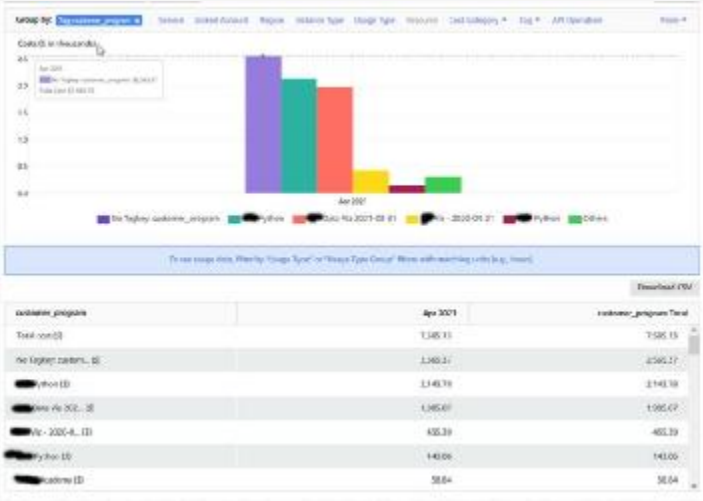
- Probably one of the most important things you can do with cloud resources is to tag them
- Just like corporations tag their physical assets with an asset tag
- Cloud resources can be similarly tagged with virtual asset tags

Key	Value
aws:cloudformation:stack-id	arn:aws:cloudformation:us-east-1:963778699255:stack/MFM
customer	[redacted]
Role	classroom
customer_program	[redacted]
Name	[redacted]
aws:cloudformation:stack-name	MFM735dda40-b6f7-11eb-a517-0fe5727aeb4a
stop_hour_utc	23
end_date	20210608
start_date	20210517
sequence_number	4

conygre training

Tags and Cost Management

- Tags can also be used to manage costs
- Reports and metrics can be generated on costs per tag

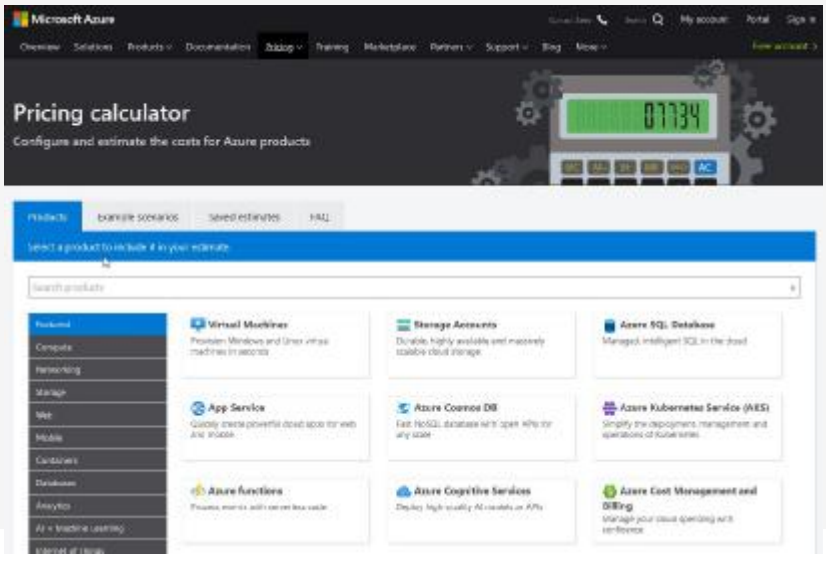


conygre training

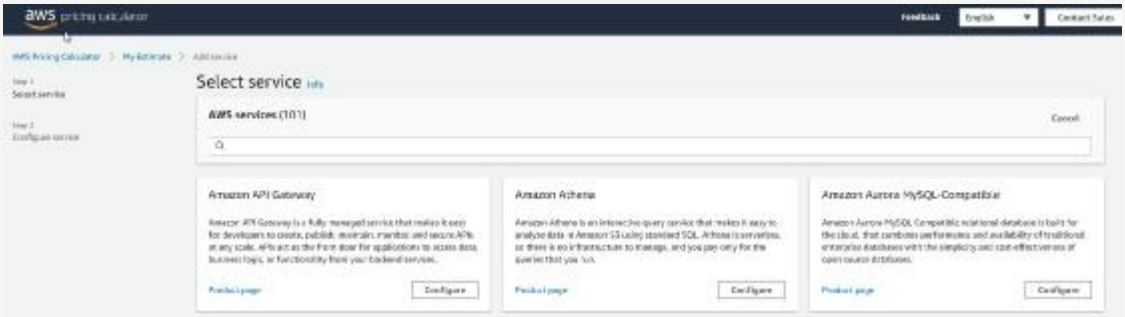
Accountability

- You need to have a culture where clear accountability is in place for costs
- Accountability can be facilitated by both
  - Tagging
  - Separate Cloud Accounts
- You will need your teams to own their own costs
- Teams can do this is if they are aware of the cost implications of what they plan to do
- The cloud providers have cost calculators to facilitate this

Pricing Calculator - Azure



Pricing Calculator - AWS



conygre training

Reservations and Spare Capacity

- Cloud Providers also offer ways to reduce your bill
  - Reserved Instances for example where you contract to have certain virtual machines for a year or three years
    - This then gives you a saving of up to 60%
- Spare Capacity
  - AWS monetize spare capacity, so you can get spare capacity on AWS at a reduced price
    - The caveat is that AWS might need it back!

conygre training

Managed Services

- Cloud Providers also offer managed services
- Might be slightly more expensive on your monthly bill, but there will be much reduced management overhead, for example
  - Managed SQL instances vs you installing your favorite database on a virtual machine
  - Managed Messaging service vs you installing your favorite messaging server on a virtual machine
  - Serverless architectures over running applications on servers

Summary

- Is the Cloud Cheaper?
- Good Cloud Governance
- Good Cloud Accountability
- Cloud Provider Utilities for Cost Management