cloudsoft
Bringing Business to the Cloud

A CLOUDSOFT WHITEPAPER

Comparing AWS and Azure

12 reasons to run Microsoft Workloads on AWS

Cloudsoft
November 2018

Table of Contents

Executive Summary

Like a bank customer that never moves their account, or an insurance customer that never finds a competing quote, it's common to see Microsoft developers and administrators default to Azure as their first public cloud without considering a different hyperscale cloud. It appears logical to think that Microsoft would be the best cloud to run Microsoft workloads: but is this true?

Cloudsoft have been working with cloud computing for a decade and currently help customers around the world to migrate, run and evolve their applications on a range of clouds. With this experience, Cloudsoft undertook some analysis of AWS and Azure to answer the question, "Which is the best cloud to run Microsoft workloads?"

The analysis included researching industry analysis, customer case studies, learning from other practitioners via events, and looking at the hard facts of cloud performance and capabilities.

Cloudsoft's research discovered that the answer to the question is this:

Keep your Microsoft developer environment and Microsoft software, but run your Microsoft workloads on AWS.

This approach offers the best of both worlds: familiar software, but running on the biggest, most mature and leading public cloud with the richest ecosystem.

In this paper, Cloudsoft explain twelve reasons for running Microsoft workloads on AWS:

1. Keep the same Microsoft Developer eXperience
2. Full .NET on AWS
3. Best tools, best cloud
4. Keep the same Microsoft software and licenses
5. Global reach and real Regions
6. Uptimes and Outages: the AZ reality
7. SQL Server High Availability: cheaper on AWS
8. 10 Years of Microsoft on AWS
9. More Microsoft on AWS than Microsoft on Microsoft
10. More Microsoft licensing options on AWS
11. Cloud Native Cloud
12. Cloud Native Partners

To discuss this report and get help running Microsoft on AWS, contact Cloudsoft.

About Cloudsoft

Cloudsoft are an AWS Advanced Consulting partner with a decade of application-centred cloud experience. Customers use our AMP and Visual Composer software to do their own application-centred cloud, or they use our migrate, run and evolve services with our help.

The Cloudsoft team is uniquely made up of cloud engineers that are experts in applications, automation and AWS. Software is in the DNA of Cloudsoft and this is a huge help to our customers, from Tier 1 global banks across to small startups, as we extend their cloud and engineering teams with expertise through our software and services.

Contact Cloudsoft about this paper.

1. Keep the same Developer eXperience

Microsoft DX (Developer eXperience) is a leading program and there are many loyal .net developers in the industry. Microsoft invested in DX because Developers are referred to as Kingmakers and therefore if they develop on Microsoft they will naturally want to deploy on Microsoft. However, this logic is flawed because Developers do not usually want to:

1. Get distracted from application development into the detail of administering and managing a hyperscale cloud.
2. Perform the due diligence required for the "big picture" analysis of choosing the right cloud for the business - for example, they *might* consider an Azure-equivalent of AWS Well Architected Framework but they are unlikely to consider, or want to be involved, anything like the AWS Cloud Adoption Framework which is something every business needs to consider to be successful.

These frameworks reveal that if a development team chooses a cloud because it suits just their immediate needs without looking at the other factors from finance to operations to security is not a recipe for cloud success. This is born out by countless case studies and stories that talk about things like culture, people and process being vital.

While it's important to "get going" and not suffer cloud analysis paralysis, it's important to put Developer needs (languages, frameworks and tools) into perspective when looking at the whole of a hyperscale public cloud.

2. Full .NET on AWS

According to CIO Magazine, 74% of Fortune 500 enterprises have line-of-business applications running on .NET. The 2017 Stack Overflow developer survey reported that C# is the third most popular server-side language and .NET Core the second-most popular server-side framework after node.js.

Why put .NET on AWS?

- No need to move your code, integration with VSTS and AWS
- Improve your security posture
- Increase innovation with AWS services to easily leverage AI
- Partnership – Microsoft Gold Partner, Premier Support
  - Ability to easily and quickly deploy .NET Core apps to Linux AMI preconfigured for you
  - Extensive support for .NET Core across all new and existing AWS Services with new services constantly being released.

Recently Launched on AWS: .NET Core 2.0 Support: Lambda, AWS Codebuild, X-Ray, AMI- .NET Core
- Lambda: functions as a service for .NET Core 1.0 and 2.0
- AWS Codebuild: fully managed build service
- AWS X-ray: debug .NET Core apps running on AWS
- EC2- new AMI pre- configured for .NET Core 2.0

3. Best tools, best cloud

The most disruptive change for a developer in the migration to public cloud is a change to the toolset. The good news is that with either AWS or Azure, the same toolset can be used - Visual Studio Team Server (VSTS, or what Azure calls DevOps).

Source:
https://www.slideshare.net/AmazonWebServices/developing-applications-on-aws-with-net-core-aws-cape-town-summit-2018

The AWS Tools for Microsoft Visual Studio Team Services (VSTS) are extensions for Microsoft VSTS and on-premises Microsoft Team Foundation Server (TFS) that make it easy to deploy .NET applications using AWS.

If you already use Microsoft VSTS or TFS, the AWS Tools for VSTS make it easy to deploy your code to AWS using either AWS Elastic Beanstalk or AWS CodeDeploy. No changes to your existing build/release pipeline and processes are required to integrate with AWS Services.

You can even deploy serverless .NET Core applications and .NET Core C# functions to AWS Lambda. The AWS Tools for VSTS allow you to deploy AWS CloudFormation templates, so you have an easy way to manage, provision, and update a collection of AWS resources from within VSTS. AWS Tools for Visual Studio provide integration with many AWS services, which make it easy to store build artifacts in Amazon S3, run commands from the AWS Tools for Windows PowerShell and AWS CLI, and manage notifications through Amazon SNS or Amazon SQS queues.

Running AWS Toolkit for Visual Studio

Source: https://docs.aws.amazon.com/toolkit-for-visual-studio/latest/user-guide/setup.html
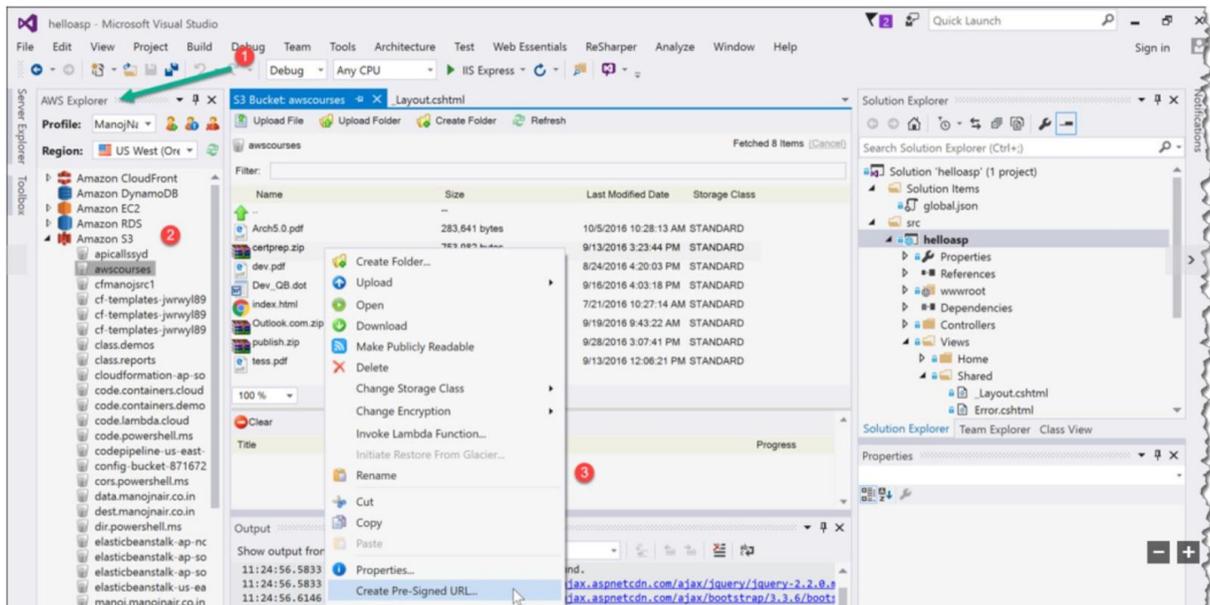
You can start using the AWS Tools for VSTS at no additional cost from the Microsoft VSTS Marketplace, or on Github.

4. Keep the same Microsoft software and licenses

Microsoft software is used by just about every enterprise one would ever meet. It's familiar. Usually an Enterprise Agreement is in place. Therefore it's the easiest thing in the world to leverage that EA to start using Azure without considering any other options.

Microsoft are counting on this kind of customer behaviour to lazily bind customers to Azure. It's a strategy similar to banking where customers leave their accounts dormant and never reviewing the interest rate, and the insurance business where consumers automatically renew and never compare quotes.

It is assumed that Azure is the best place for Microsoft workloads. But, like most things in life, the truth is counter-intuitive: you can use your existing Microsoft software and licenses on AWS, plus you can rent additional licence-included instances without having to modify your agreement with Microsoft.

5. Global reach and real Regions

Global reach matters for many business today. Scottish businesses serve international customers as far afield as New Zealand. Running an application in a datacenter in Glasgow serving customers in Auckland is known to be a poor experience due to the latency of the

network. This is one compelling reason that a business chooses the public cloud: to improve customer experience by running workloads or delivering assets local to the customer.

AWS and Azure both have enormous global networks and local-to-the-region deployments of cloud resources. Caution when comparing: comparing AWS Regions to Azure Regions is like comparing apples to oranges.

- An AWS region consists of multiple Availability Zones owned and operated by AWS (except for China). An Azure region can be one datacenter or co-location space in a third-party datacenter.
- Azure has only recently deployed Availability Zones, in one Region
- Azure doesn't always own and operate all of its own data centers, which is one reason it has been able to "overtake" AWS in number of "regions". Microsoft are known to use datacenter partners like Equinix in Germany and London.

See [Microsoft Sep 2018 post-mortem for VSTS/DevOps](#) for acknowledgement of this ("Moving forward, we plan to address failures within a region by using Availability Zones").

6. Uptime & Outages: the AZ reality

Microsoft have acknowledged that Azure doesn't have a comparable implementation of the AWS Availability Zones -- it's a fundamental design issue recognised by Microsoft, and the industry, and it has contributed to a [recent Sep 2018 outage on VSTS/DevOps](#).

> *Gartner analyst Lydia Leong has recommended considering disaster recovery capabilities away from Azure for critical applications hosted in the cloud.*

An AWS Region consists of two or more Availability Zones, each AZ is made up of one or more data centers. Typically an AWS Region has three AZs, some like those in the US have even more. AZs are mesh connected together with high-speed, high-bandwidth, low-latency private network connectivity.

AWS AZs are not only *crucial* for their high uptimes -- if the application can exploit the architecture, which is not a given -- but AZs are *essential* for reducing SQL Server licensing costs because the AZ connectivity latency of sub two milliseconds enables synchronous SQL Server high availability. This can save from thirty to fifty percent off SQL Server licensing costs (tens of thousands of pounds in small cases).

The Azure move to an AZ model is welcome, and is available today in one region, but it is new and rolling this out will take billions of dollars and some time. Customers should buy what is here and now, they shouldn't buy a promise or a roadmap.

> *"Beyond the task to build and connect more data centers, Microsoft must also modify its software to accommodate a multi-availability-zone architecture," said Lydia Leong, Gartner Analyst and author of the IaaS Magic Quadrant.*

AWS are not impervious to outages. See this detailed analysis by Thousand Eyes for details of a March 2018 AWS outage.

7. SQL Server High Availability: Cheaper on AWS

The AWS Availability Zones in each Region are connected via networks that give sub-two-millisecond latency which is fast enough for synchronous data replication. This means you can use a cheaper SQL Server architecture and licensing option to save around a third of your SQL Server licensing, and deploy a simpler solution.

Also see TSO Logic TCO Analysis:

"One TSO Logic customer was using 1,356 licensed cores of SQL Server on-premises and wanted to bring their SQL license to AWS. Previously, they would have had to use 2,220 cores (63 percent more) on AWS to meet their memory needs.

Considering the list price for Microsoft SQL Server is $14,000 per core, that fixed core count requirement would have meant a huge spike in licensing costs—effectively making cloud a nonstarter for this workload.

Now, customers can keep the same memory, storage, and bandwidth of the full-size instance they need. They can also run SQL Server on AWS with just 1,272 cores—7 percent fewer than they're currently using on-premises. Running on AWS with 84 fewer cores translates to a significant price drop over this customer's current environment.
More importantly, the core count flexibility means the AWS Cloud is now a viable option for their SQL Server. They can bring their existing license and start realizing all the other designed benefits of the AWS Cloud."

8. Ten Years of Microsoft on AWS

Andy Jassey, CEO of AWS, once famously addressed the question "Are Microsoft catching you up?" with the answer "There is no compression algorithm for experience."

AWS have been running Microsoft workloads for ten years, which is the lifetime of Azure.

> Celebrating 10 years of Microsoft Windows Server and SQL Server on AWS! Happy Birthday!

Customers running Microsoft workloads on AWS, and why:

1. **Ancestry** - *"AWS provides us with the flexibility we need to stay at the forefront of consumer genomics, as the science and technology in the space continues to rapidly*
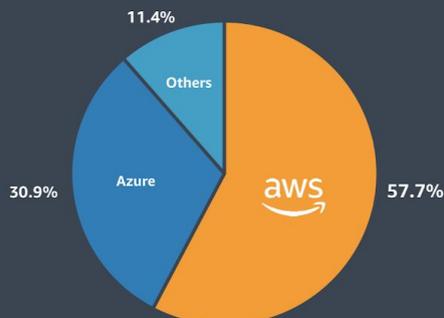
evolve. We're confident that AWS provides us with unmatched scalability, security, and privacy."

2. **eMarketer** - *"We chose to move our Microsoft workloads to AWS because of your extensive migration experience, higher availability, and better performance. We are seeing 35% cost savings and thrilled to see 4x faster launch times now." – Ryan Hoffman, Senior Vice President of Engineering*

3. **Agero** - *"We experimented with AWS Elastic Beanstalk and found it was the simplest, fastest way to get .NET code running in AWS." – Bernie Gracy, Chief Digital Officer*

9. More Microsoft on AWS than Microsoft on Microsoft

Source: https://d1.awsstatic.com/analyst-reports/IDC_Slide_WindowsonAWS_JM181015.pdf

AWS runs more Microsoft on AWS than any other provider - by 2x - IDC (a leading IT Analyst) estimates that AWS accounted for approximately 57.7% of total Windows instances in public cloud IaaS during 2017 – nearly 2x the nearest cloud provider.
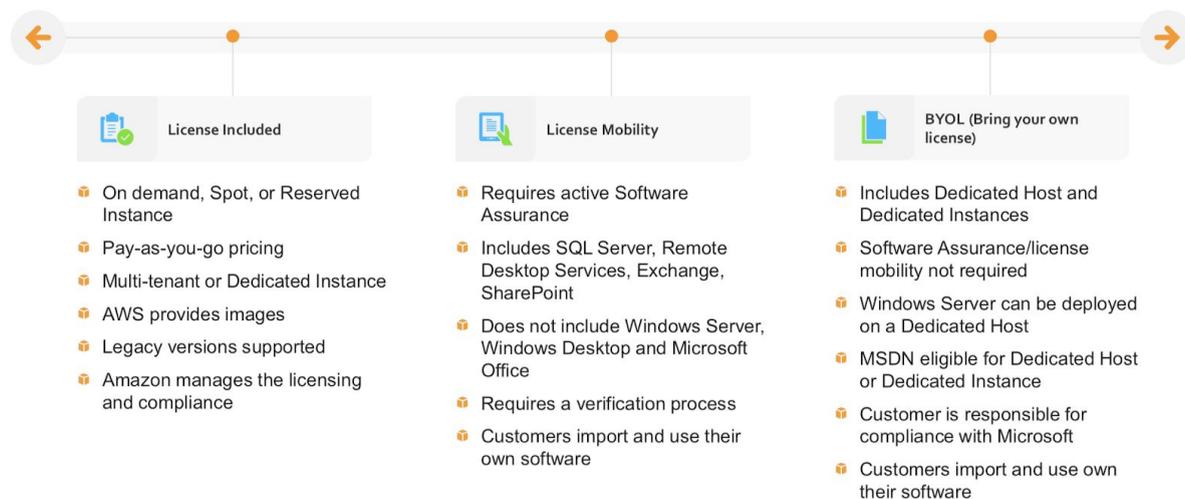
All Microsoft workloads run on AWS:

● Windows Server: AWS supports many Windows Server editions (2008/R2, 2012/R2, 2017/Base/Containers, and even Windows Server 2003)
● SQL Server: AWS supports many versions of SQL Server (2005, 2008, 2012, 2014, and 2017RC) in classic VMs (EC2) or our managed Relational Database Service (Amazon RDS)
● Languages: AWS can run all .NET CLR languages on EC2.

- Serverless: AWS Lambda supports .NET Core 1.0 and 2.0 (in addition to Node.JS, Python, and Java).

Licensing: With License Mobility and Software Assurance, customers can leverage existing licenses through Bring Your Own License (BYOL) or utilize Pay-As-You-Go (license-included) options.

10. More Microsoft Licensing options on AWS

There are three options -- and these can be mixed and matched to give maximum flexibility to the business when running Microsoft on AWS:

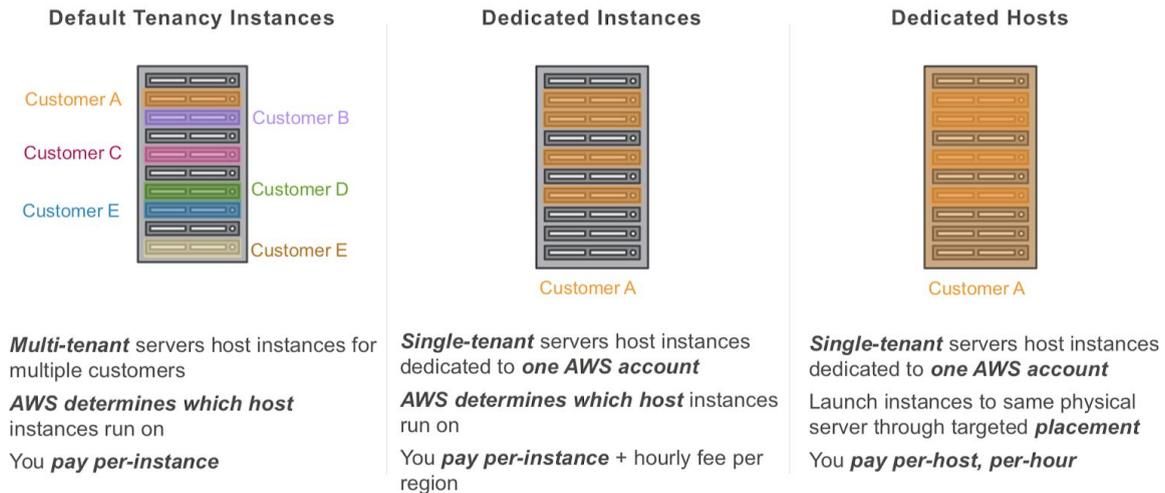| License Included	License Mobility	BYOL (Bring your own license)
On demand, Spot, or Reserved Instance	Requires active Software Assurance	Includes Dedicated Host and Dedicated Instances
Pay-as-you-go pricing	Includes SQL Server, Remote Desktop Services, Exchange, SharePoint	Software Assurance/license mobility not required
Multi-tenant or Dedicated Instance	Does not include Windows Server, Windows Desktop and Microsoft Office	Windows Server can be deployed on a Dedicated Host
AWS provides images	Requires a verification process	MSDN eligible for Dedicated Host or Dedicated Instance
Legacy versions supported	Customers import and use their own software	Customer is responsible for compliance with Microsoft
Amazon manages the licensing and compliance		Customers import and use own their software

Source:
https://www.slideshare.net/AmazonWebServices/best-practices-for-bringing-microsoft-license-and-applications-to-aws-by-sean-lewis

Microsoft software available via License Included:

- Windows Server (2003,2008/R2,2012/R2,2016,2017)
  - Includes two administrative Remote Desktop Services (RDS) connections (additional RDS licenses can be brought through license mobility)

- SQL Server (2005, 2008, 2012, 2014, 2016)
  - Standard, Enterprise, Web Editions
  - Per-core, minimum 4, no CAL needed
  - Amazon EC2 or Amazon RDS

- Pricing models
  - On-Demand (per hour)
  - Reserved Instances (1 year, 3 years, with or without upfront)
  - Spot Instances

Choosing a license model for software that runs on EC2 also means choosing one of three tenancy models depending on license choice:
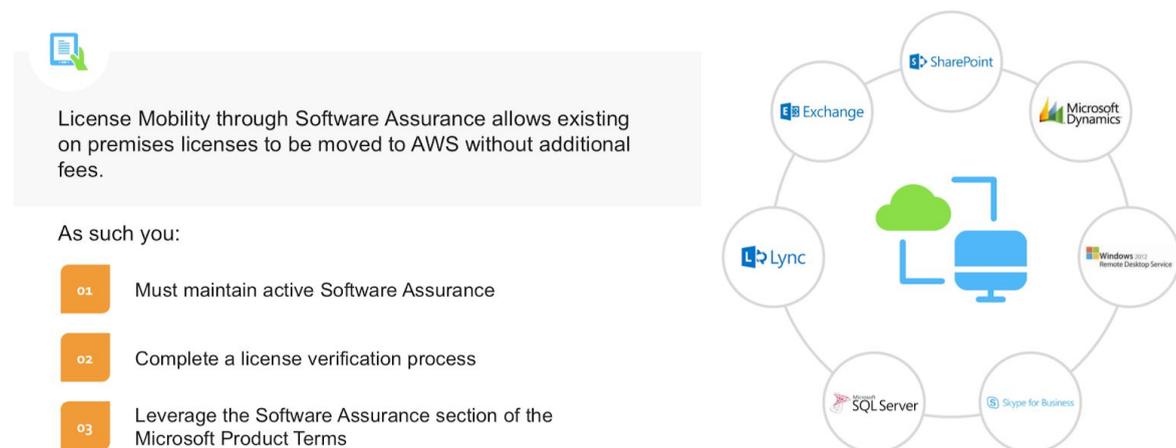

EC2 Instances: Default and Dedicated Tenancy

Default Tenancy Instances

Multi-tenant servers host instances for multiple customers

AWS determines which host instances run on

You **pay per-instance**

Dedicated Instances

Single-tenant servers host instances dedicated to **one AWS account**

AWS determines which host instances run on

You **pay per-instance** + hourly fee per region

Dedicated Hosts

Single-tenant servers host instances dedicated to **one AWS account**

Launch instances to same physical server through targeted **placement**

You **pay per-host, per-hour**

Source:
https://www.slideshare.net/AmazonWebServices/best-practices-for-bringing-microsoft-license-and-applications-to-aws-by-sean-lewis

It is also possible to use License Mobility through Software Assurance:

License Mobility through Software Assurance allows existing on premises licenses to be moved to AWS without additional fees.

As such you:

01 Must maintain active Software Assurance

02 Complete a license verification process

03 Leverage the Software Assurance section of the Microsoft Product Terms

Source:
https://www.slideshare.net/AmazonWebServices/best-practices-for-bringing-microsoft-license-and-applications-to-aws-by-sean-lewis

Specifically for SQL Server and License Mobility:

- SQL Server is licensed by vCPU rather than physical core

- Customer brings a number of cores equal to core counts on table to launch specific instance sizes.
- Four Core minimum per instance
- SQL Passive failover benefits enabled
- This is a small subset for example purposes. For full list, please visit:https://aws.amazon.com/windows/resources/licensemobility/sql/

11. AWS is a Cloud Native Cloud

The AWS cloud started life in 2006 as a response to the Amazon CTO, Werner Vogels, frustration with "traditional, legacy" software and hardware providers. AWS engineers went to South Africa and built a Amazon retail a new technology answer from scratch: the AWS cloud was born and has been cloud native from the start and the bottom-up. AWS leverage a lot of Open Source Software and their cloud is customer built to be a cloud with no legacy.

Conversely, Microsoft were dragged kicking and screaming into the cloud because their business was threatened by the cloud. Microsoft have to use and modify existing, non-cloud software like Windows Server/HyperV and hardware such as commodity kit from Dell, to create their first cloud systems.

Since those early days, under Satya Nadella, Microsoft have fully embraced open source (even buying Github) and they are the #1 contributor to Open Source in the world. However, the technology in their clouds is still based on the non-cloud Windows Server, Hyper-V and they continue to have to invest billions of dollars to evolve that legacy technology to keep pace with the next generation. One significant example is their current goals of moving to the AWS Availability Zone model, the lack of which is a weakness in Azure.

12. AWS has more Cloud Native Partners

Both Microsoft and AWS have enormous partner programs but they are very different.

The Azure partner ecosystem is largely made of previously non-cloud partners who are MVP (Most Valuable Partners - specialists) and Channel Resellers. These partners are experts in the non-cloud world but have had to evolve to be cloud relevant. They *generally* don't have application skills or cloud native skills unless they have been built that way from day one.

There are specialist Microsoft partners who are cloud and application experts, but they are not the majority. Microsoft customers must beware of organizations that have "bolted on" an Azure practice to a non-cloud business. This also extends to Microsoft-certified Azure engineers who are often "bottom-up cloud plumbers" who know their way around the Azure Resource Manager but do not have the skills higher up with applications, databases or other services.

The AWS partner ecosystem also has these kinds of partners, non-cloud MVPs and channel resellers, but it also has -- because of its original cloud native history -- a lot of startup-related partners and individuals who have lots of "higher order" skills, above cloud plumbing, in application and databases, plus advanced serverless, AI and ML. Simply by being the longest-living and biggest cloud, there are more partners with deeper, wider and higher cloud skills.

Cloudsoft is an example of a cloud-native, application-centred AWS partner.

Crisp Report: A comprehensive Cloud Service Provider analysis framework

Full report available from the AWS Analyst page - see Crisp Research: Evaluation of Cloud Computing Providers, September 2018. German analyst house Crisp Research confirms AWS' position as a leading cloud platform in their new vendor evaluation.

Cloud Services and Products

Features / Service Portfolio - AWS 95% vs Azure 80%
- Infrastructure Services Platform Services Architecture Options Feature Set
- Hybrid & Multi Cloud Capabilities
- Portfolio Completeness
- Service Capabilities & Service Management Vendor Selection
- Management Services
- Technology Selection & Portfolio
- Platform Independency

Service / Product Experience - AWS 90% vs Azure 70%
- Service & Support Design
- Set Up & Onboarding
- Infrastructure Performance & Connectivity Operational Model
- Deployment Models
- Workload Variety
- Value-Adds (e.g. Management-Tools, cloud-native Development)
- Deployment Options & Data Center Value-Adds (e.g. Monitoring, Portal, Dashboard)
- Service & Process Design
- Admin Experience
- Testing Capabilities

Integration - AWS 90% vs Azure 90%
- APIs & API Documentation Management Hybrid & Multi Cloud Capabilities Integration Capabilities
- APIs

- Public Cloud Integration
- Templates / Blueprints
- Integration Experience Multi-Cloud-Integration
- APIs / Hybrid-/ Multi-Cloud-Integration Deployment Models
- Data Protection & Security Management

Economics - AWS 80% vs 75%
- Cost & Capacity Management Pricing Model
- Business Model

Disruptive Potential - AWS 80% vs 80%
- Next Generation Technology Portfolio New Business Creation
- Global Availability Zones
- AI & Automation Services
- Additional Customer Value Add Process Optimization Infrastructure & Optimization New Business Creation Business Consulting
- Digital Transformation Excellence Own Technology Portfolio

CSP Capabilities

Market Understanding - AWS 90% vs Azure 80%
- Focus
- Thought Leadership
- Focus (Pure Play MPCP)
- Company Strategy & Core Business

Footprint - AWS 85% vs Azure 70%
- Market Awareness & Visibility Regional Presence
- Reference Customers
- Regional Go To Market Strategy Enterprise IT Footprint
- Community Engagement & Alliances

Ecosystem - AWS 85% vs Azure 90%
- Certifications
- Number & Quality of Partners Managed Service Provider Partnerships Partner Landscape (Number & Quality, Enablement)
- Container Network
- Certifications (Provider & Employees, Type & Number)
- Partnering
- Partner Status at Cloud Providers

Customer Experience - AWS 90% vs Azure 75%
- Managed Services & Support Quality

- Onboarding & Price Transparency
- Customer Feedback & Satisfaction
- Digital Customer Experience Onboarding
- Container Engineers & Developers Service & Support Quality
- Education & Trainings
- Cloud Engineers & Developers

Agility - AWS 85% vs Azure 80%
- Market Responsiveness
- Innovation Budget
- Innovation Capabilities
- Release Management
- Agile Development / Release-Management Influence-potential on Vendors