MTC Learnings from ISV and Enterprise engagements

Govind Kanshi
About me

- @govindk
- [http://govindkanshi.wordpress.com](http://govindkanshi.wordpress.com)
- Databases and applications is the focus.
- MTC India
Agenda

• Most common issues
• Lift & Shift – or start blaming everybody else
• DR & Backup – there is no clustering?
• Performance – why is my disk so slow
• Network – what does a CIDR mean
• Some services – what they do and how you can use them
Migration issues (top issues we get)

- Sticky session (ARR) – fixed now (use ps command to create tuple)
- Isolation (machine should not go out of subnet) – fixed
- Multiple Ips/NICs - fixed (NICs fixed, IP coming)
  - Management NW
- Disk performance
  - Provisioned (fixed)
  - SSD (fixed)
- OS – X – need exception talk to vendor – Talk to vendor
- Oracle/SAP/DB2 – need to go for support from them
- No multicast allowed - Java based App servers can use JGroups
- SNMP not present – in most public clouds
Practical issues

• Issue (Operations)
  • Sprawl of subscriptions, VMs (monitor)
  • Running out of core, storage accounts or skewed account usage - monitor
  • Granular billing (+ tags - coming )
  • Better Security mechanism (RBAC getting there)
  • Run out of Network (properly allocate CIDR)

• Naming conventions
  • Name_of_proj_imageName_purpose_region (no need of tag)
Goals differ for ISVs and Enterprises
<table>
<thead>
<tr>
<th>ISV</th>
<th>Enterprise</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agility for change</td>
<td>Stability with some agility</td>
</tr>
<tr>
<td>Shared Capex</td>
<td>Shared Capex across stakeholders</td>
</tr>
<tr>
<td>SaaS</td>
<td>Maintain balance (old data, old systems)</td>
</tr>
<tr>
<td>Elasticity depends on customer</td>
<td>Elasticity well defined for workloads – in general.</td>
</tr>
<tr>
<td>Cost/Margins are big factor</td>
<td>Established firms know costs of people/sw and optimize</td>
</tr>
<tr>
<td>Provisioning</td>
<td>Provisioning with control</td>
</tr>
</tbody>
</table>

Need to exploit cloud infra to gain efficiencies around cost
Life is full of surprises
Lift and test - Enterprise

• Issue – In my DC/Colo/…..

• Resources are throttled in public cloud
  • Storage – throttled - You can catch Storage throttling
  • Your network bw is throttled so as to be nice to others.
  • Your vm cpu is throttled so as to be nice to the neighbor.
  • Services are throttled(shared resources)
    • Exception is o365 – dedicated client or
    • You go for largest machine (compute)

• Mismanaged expectations
  • OS support, vendor support, network, storage IOPs requirement
  • Special clustering requirement for HA
“Forklift” – with care

• Challenge is applications are very deeply integrated with each other
## Decision matrix

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data size,</td>
<td>Retire, DoNot Migrate, Replace with SaaS(work commercials), Optimize (refactor, utilize cloud offerings), Lift and shift (weigh in approaches)</td>
</tr>
<tr>
<td>Adaptation to cloud cost (storage/nw/monitoring) Badly performing app on-premise will perform worse on the cloud</td>
<td></td>
</tr>
<tr>
<td>Security implications- store data outside, auditing req</td>
<td></td>
</tr>
<tr>
<td>Workload complexity – comes with biztalk and mq series and solaris/sgi app</td>
<td></td>
</tr>
<tr>
<td>Availability -Nothing like availability sets is present on-premise</td>
<td></td>
</tr>
<tr>
<td>Location people will access apps from x</td>
<td></td>
</tr>
</tbody>
</table>
Security
What does Azure provide

• Security Development Lifecycle (SDL).
• Operational Security Assurance (OSA).
• Assume Breach.
• Incident Response
• 24 hour monitored physical security.
• Monitoring and logging.
• Antivirus/Antimalware protection.
• Intrusion detection and DDoS.
• Zero standing privileges.
• Encrypted communications.
• Penetration testing ........

• ISO/IEC 27001:2005
• SOC 1 and SOC 2 SSAE 16/ISAE 3402
• Cloud Security Alliance Cloud Controls Matrix
• FISMA
• FedRAMP
• PCI/DSS- I
• United Kingdom G-Cloud
• HIPAA
• Life Sciences GxP
• FERPA
• FIPS
Security & isolation

• Isolate using virtual network/subnet – **always use vnets to host**
  • Create proper subnets
  • Use network acls
  • Use network security groups, ACLs, firewalls

• Other services
  • SQL Azure – connection string
  • DocDB/Search - Keys
  • Storage- SAS/Regenerate keys and the list goes on

• Others
  • Use AD accounts, MFA
Security

• All connection endpoints (gateway/network permissions)
  • Who manages them
  • Who uses them
• Traditional monitoring (SNMP) does not work
• Data at rest encryption is your responsibility (for now)
  • SQL Azure...
  • SQL Azure has auditing too
  • Do your own on SQL on VM or storage
• Key management is an issue – have process of attribution and checks
  • RBAC across services – starting
  • Auditing – log available
Availability

• Issue
  • On premise we use cluster of some kind
  • We do not think of Datacenter/Racks today
    • Our admins do that

• DB on VMs
  • SQLServer – Always ON (don’t compromise availability for cost )
  • Oracle – DG, ADG, GG
  • MySql – master slave, NDB cluster
  • Mongo – master slave

• Look at every service availability (it varies)
Ref - Compute

Availability

- Notification of downtime
  - No single machine SLA – Availability group with at least 2 instances
  - Need to work on SLA by replicating data and settings
  - Generally 2 pair of app + db works fine
  - Cache etc require re-building
  - VPN connectivity availability

- DNS/NW
  - Use 3rd party DNS
  - VPN connectivity availability vs expressroute

- Services (example)
  - Redis cache
    - Master Slave (auto failover – hopefully more transparency in future)
  - SQL Azure/Queue/Storage
    - 3 replicas + RO + geo replication (Where applicable)

- Monitor from external endpoints, inside apps, inside Azure
- Think about availability at all levels
Availability sets

• Compute need to be in availability set
  • Some workloads do not enable themselves for Availability set

• Plan for DR in another region by
  • Pushing configuration changes
  • Pushing data changes using data tech
  • Pushing cache – invalidation
  • Traffic manager is great but backend data needs to be in sync
Availability

• Test it (develop your own chaos monkey)
  • Hosted services do not have failure mode so you need to go back
    • Kill the connection or connect to wrong/unexisting machine.
  • Measure everything – tools time, data restore time, verification time, people interaction time – literally have a log book which keeps improving over time to include other events
  • Use hysterix and similar approaches – circuit breakers to overcome service issues
  • Canaries across the services (applies to perf too)
Performance
Performance

• What is better D or A series –
  • Do the test
    • Cpu/io at least

• Choose right vm – try scale out & scale up
  • It all depends – DBs like scaleup

• Reiterate - Choose right storage
  • Local disk, SSD, ephemeral disks, shared, and persistent disk from Azure blob
  • Provisioned vs standard

• Standard-decoupling-scale-individual-pieces (SDSIP)
  • DB – scale up/R-W-Shard
  • Session Data – cache/nosql or chose right store
  • Front end assets – use CDN, use varnish
  • Load balancer – Internal-External or nginx, HA proxy
  • Auto scale — plan for it and test it
Do not forget basics

- Use perf tools
  - NW – iperf
  - Disk – iozone
  - Memory – stream

- Load balancer
  - You don’t have control over size/notifications (in a way good)
  - Myth - LB is ROUND ROBIN - nope
  - Operations - No logs yet, can’t install monitoring agents or see the stats (coming)
  - Operations - SSL termination does not happen on LB (coming)
Performance – things you will find

• NW
  • Machines have BW barrier – which keeps going up
  • NW gateways have barrier – 200 Mbps
    • Even though internal nw could be GB hookup
  • For enterprise scenarios
    • Location based pipes to VNETs (use express route)
  • Use New regional VNET ensures assets are close by
  • Use New SQL image pre-striped with storage pool available for SQL Transactional workload
Performance

• Monitor
  • Reachability, latency, throughput
  • Within app telemetry – newrelic/appinsight/erroception for js etc
    • Latency
  • App –stack monitoring
    • OpsInsights or agent based sw – boundary/scom/datadog etc...
    • Perfmon counters, error logs, app logs
    • Monitor logs – error/syslog – logstash is simplest but ymmv
    • Collectd/StatsD + fav collection tool(flume to x ) + visualization graphite to x – identifying issues
  • Monitor services
    • Request for API based pull of data so that your “app” can have 360 view
Save Money

• Issue
  • Ran out of budget in days/weeks/months (ran large machine)

• Other side of pay as you go
  • You pay even if you do not use but keep services on

• Do custom provisioning and de-provisioning to take care of growth and lag - you need to think through “quiecising”

• Think through excessive disk space usage – you pay by “storage”

• Switch off unused/unwanted vm instances and orphan storage disks
Exploit azure to get cost efficiencies

• Exploit Azure
  • Don’t just move compute and storage
  • It requires rework on part of software
    • Can I do without full fledged relational db
    • Can I use pre-generate reports and store them in low cost storage
    • Can I use smaller machines
    • Can I start using lower cost services for search/cache/json or nosql store

• Look at long term (3 year) for ROI –
  • Azure EA(if you have SA- you will have lot of sleep) is great steal
  • Don't forget your hvac, real estate, people, rent, provisioning, cost of DR-HA, licensing

• Look at agility and the cost of not having it
• Always get Azure support - it is small price to pay for the peace
• Trust but validate
Your Feedback is Important

Fill out evaluation of this session and help shape future events.

**OPTION 1**

Scan the QR code to evaluate this session on your mobile device.

**OPTION 2**

You can fill out evaluation of this session directly through the App.

**OPTION 3:** Feedback stations outside the hall
## Rough guide

<table>
<thead>
<tr>
<th>Category</th>
<th>Storage</th>
<th>Compute</th>
<th>CDN</th>
<th>LB</th>
<th>Monitoring</th>
<th>Data</th>
<th>DW</th>
<th>Ingestion, Integration and Messaging</th>
</tr>
</thead>
<tbody>
<tr>
<td>NW</td>
<td>MPLS, VPN</td>
<td>MPLS, VPN/Expressroute, VirtualNetwork (dynamic advertisements of routes coming)</td>
<td>SSD/Voilin/NAS/San/Das</td>
<td>Local/SSD/Ephemeral/VHDs from storage, availability/rr/geo</td>
<td>Raw/vm on hyperv/vmware/amzn</td>
<td>Inmage tool to convert, Azure Iaas or PaaS</td>
<td>CDN</td>
<td>CDN</td>
</tr>
<tr>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
<td>CDN</td>
</tr>
<tr>
<td>LB</td>
<td>F5, custom-sw</td>
<td>External Load balancer, Internal, run your own</td>
<td>F5, custom-sw</td>
<td>F5, custom-sw</td>
<td>F5, custom-sw</td>
<td>F5, custom-sw</td>
<td>F5, custom-sw</td>
<td>F5, custom-sw</td>
</tr>
<tr>
<td>Monitoring</td>
<td>Scom, Nagios or just tail log file</td>
<td>Scom, new relic, boundary, gomex, keynote, Nagios, cacti, Azure metrics (Paas/Iaas – linux coming)</td>
<td>Scom, Nagios or just tail log file</td>
<td>Scom, Nagios or just tail log file</td>
<td>Scom, Nagios or just tail log file</td>
<td>Scom, Nagios or just tail log file</td>
<td>Scom, Nagios or just tail log file</td>
<td>Scom, Nagios or just tail log file</td>
</tr>
</tbody>
</table>
## Rough guide

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CEP</strong></td>
<td><strong>Streaminsight</strong></td>
<td><strong>Streaming Analytics</strong></td>
</tr>
<tr>
<td><strong>Batch Jobs</strong></td>
<td></td>
<td><strong>Azure Automation</strong></td>
</tr>
<tr>
<td><strong>Caching</strong></td>
<td><strong>memcache, appfabric, redis</strong></td>
<td><strong>hosted redis, document db</strong></td>
</tr>
<tr>
<td><strong>Identity - AD -</strong></td>
<td><strong>AD</strong></td>
<td><strong>Azure AD (EMS)</strong></td>
</tr>
<tr>
<td><strong>RMS</strong></td>
<td><strong>RMS</strong></td>
<td><strong>Azure RMS (EMS)</strong></td>
</tr>
<tr>
<td><strong>Management of assets</strong></td>
<td><strong>Intune, System Center</strong></td>
<td><strong>Intune (EMS)</strong></td>
</tr>
<tr>
<td><strong>Access to apps on byod</strong></td>
<td><strong>EMS</strong></td>
<td><strong>EMS</strong></td>
</tr>
<tr>
<td><strong>Backup -</strong></td>
<td><strong>Tapes, custom SW</strong></td>
<td><strong>Azure StorSimple, Backup Vault</strong></td>
</tr>
<tr>
<td><strong>Monetization/Mobility -</strong></td>
<td></td>
<td><strong>Azure Mobile service/API management</strong></td>
</tr>
<tr>
<td><strong>Dynamics/CRM</strong></td>
<td><strong>On-premise</strong></td>
<td><strong>On-Azure or Hosted</strong></td>
</tr>
<tr>
<td><strong>ES/Solr/Caching</strong></td>
<td><strong>On premise</strong></td>
<td><strong>Hosted Azure services for redis/search/DocDB</strong></td>
</tr>
</tbody>
</table>
Services

• Always plan for “moving out”
  • Your own datacenter, co-lo
  • Applications have some abstraction layer to plug in services
    • Storage for example – plug in behind at least an interface to allow “