# Microsoft Technical Summit 2017 | Bangalore | Social: #MSTechSummit

## Day 1

### **Key note | Corey Sanders**

Three ambitions: Create more personal computing, Reinventing productivity and business processes, Build the intelligent cloud platform.

Important cloud capabilities: Productivity, Business Apps, Application innovation, Data and intelligence, Security and management.

38 Azure regions, more than 1 datacenter in many of those regions; covers 53 compliance offerings.

Hybrid is about consistency. More than 90% Fortune 500 orgs use Microsoft cloud.

User experience

Narayana Health | Azure migration from managed hosting facility. Done in 3 months, End to end monitoring of services and applications, including that of Azure own. Guiding principles: Affordability of solutions | Balance performance, cost, and quality | Necessity driven solutioning.

Azure is open in terms of platform services and core infrastructure.

Hybrid management and security: Azure monitor | Security Center.

Developer tools: VS, VS Code, VSTS, Xamarin. Platforms services: Web-Mobile, Microservices, IoT.

Security response demo: Security incident management app. with Azure search integrated; phonetic search options are available alongside regulars like GSM, citizen database, etc., Azure elastic DB pool provides unified view of all the databases hosted in Azure DB PaaS. Manage and monitor databases from a single console. Shard queries are possible, with data spread across multiple databases in the pool. PowerBI dashboard based reporting, inherent, for a high level overview of the events and incidents across the monitored-managed area. Incident themes helps know the major event-incident reasons. This data could then be fed into Azure ML Studio, to perform data modelling and generate outputs on the events that are likely in the future. This solution can seamlessly integrate with mobile devices, for on-the-job action and updates. VSTS is a complete DevOps solution.

TalkTalk: Azure Service Fabric + Media Services | Alaska case study | Genie case study.

Mobile office solutions: Windows 10 Enterprise | Office 365 | Enterprise Mobility + Security

Office 365 mail service: contacts addressed automatically (in To) as soon as they are mentioned, Archiving emails using swipe actions, Scheduling meetings with couple of flips on the mobile phone, Microsoft teams - cloud based meeting service, Word, Powerpoint: Enhancements to search for topics and present-ability.

### **An overview of the Microsoft Application platform for Developers | Aniruddha Deswandikar**

Developer opportunity in modern business: From functional apps to Amazing experiences

Developer opportunity: Delivers rich web and mobile apps. From Isolated apps To Apps connected to the business Developer opportunity: Extend business to the cloud.

Old models do not work anymore.

Azure Enterprise Cloud Platform: Create rich mobile applications, extend your applications to cloud

Enterprises are going mobile on multiple platforms.

The mobile app dilemma: End user experience vs. Better TCO, productivity, and development experience. Time to develop is critical. There is a native and hybrid development model, and a mid-path must be found to honor user experience and time to develop. C# has best of both the worlds, with inclusion of Xamarin. 2 to 3 layers of common code wrote just once, and other areas such as UX can be split using platform specific modules.

Mobile development with .NET: 10-12% of the code is native, with Xamarin development platform.

A/B testing: Alternative UX testing on select group of consumers-users before rolling out the changes onto the production for general use. This is quintessential, with extremely short software delivery cycles.

VS Mobile Center: Azure mobile center SDK shall be subscribed to and integrated with our native development platform, for build and test.

The cloud journey: IaaS - On-premise servers | Cloud infrastructure, PaaS - Born in the cloud; Microservices.

Business models drive growth; Snapdeal selling cars alongside insurance tied to select companies is an example.

Forrester favors PaaS on the ground of Total Economic Impact results.

Azure application platform: Functions | App Service | Service Fabric | Azure Container Service, from lightweight to heavy weight application development platforms/services with degrees of isolation.

Azure Functions Demo: Resource Groups are the way to manage resources and services as kind of an atomic unit, for the purposes of billing, security, upgrade, etc.

Consumption Plan for budgetary implementations and App Service Plan for full-fledged deployments for real business purpose; the former could be relatively cheaper than the latter.

Azure Functions: C# and Javascript base available; others shall be included shortly. Functions can be tied with AAD or third party IDM services.

New application model: Core systems must run lead footprint for what is necessary and ancillaries pushed to the top with the help of microservices, and asynch data streams flowing back into these core systems. Azure Service Fabric’ real-life use case: Airline kiosks at airports, with different application modules tied to the respective airlines delivering varied and unique functions. This kiosk system is made effective, by leveraging the Azure Service Fabric. The scaling of different micro service modules according to the dynamic business requirements helps save cost and improve system efficiency.

Journey to the cloud: Lift and Shift | Sliced PoCs of select applications in the portfolio | Migrate to PaaS

All common tools heavily used in the market are imbibed into Azure, to save time and enable cloud scale solutions. For example, Dockers is included into ACS and Visual Studio. Code quality can be built into the deployment.

Microsoft Graph: automated MFA introduced based on spam detection using rich algorithms. This interface is integrated with all of the Microsoft cloud platforms and services. AAD with MFA is a safe bet.

Localization: Pages from a region and data from another could be unproductive and inefficient, unless compliance norms or company policy demands it/mandates it.

Microservices use case: IoT event hub | Function calls from Mobile App. Event hub is a fast ingestion engine, but the volume of data has to be used by applications, which sometimes could be heavier. Azure stack can help work with or replace hosted apps.

### **Deliver high scale and low cost solutions with Azure Tiered Cloud Storage | Vamshidhar Kommineni**

Current storage challenges: Scalability limits | Risk Management. | High costs

Data leaks is a major security threat.

Azure storage: Limitless scalability | Trusted (DDOS protection, IPS/IDS, etc., at cloud scale) | Cloud economics, producing business agility.

Foundational building block of Azure: All services depend on Azure storage. Hyperscale | Durable | Secure | Highly Available | Open (Platform agnostic) | Hybrid (GA this year; 100% API consistent storage solutions)

Azure unified storage system services: IaaS: Disks | Files, PaaS: Objects | Tables | Queues

Storage durability: Strong consistency, checksums/CRC, and background scrubbing to preserve data integrity. Bit rot and data consistency over time is eradicated due to continuous re-reads due to the implementation of the cloud scale replication system.

Object storage: Unstructured | Can be huge in capacity and big numbers | Infinite scaling maintained based on sophisticated data storage modelling. Limitless scale, globally accessible, cost efficient.

Application data scenarios: Live data repository | Active or Deep Archive | Big Data Analytics.

Customer use cases | [carmax.com](http://carmax.com/) | Car rental: Original Vehicles Images Architecture, images quality and dynamic re-size-ability was at the apex of business, from selling point of view. Azure blob storage promised to deliver the solution, RA-GRS enabled for active-active load-balanced traffic management across geos, application reverse proxy done through Azure Traffic Manager, via Express Route. Rich photos | Cultural shift for teams.

ADM PoV - Azure storage concepts: Subscriptions, Storage Accounts, Containers, Blobs.

Object storage systems are Key-Value systems, with object names being the key and their data being the value.

Blobs: Block | Append (Big data) | Page. ETag helps identify data changes without peeking into.

RA-GRS: Storage client SDKs take care of the redirection from primary to secondary and need not be manually coded.

Uploading block blobs: PutBlob | PutBlock/PutBlockList. Blobs shall be deduced and can be referenced several times and charged only for unique blobs (blocks), upload only the changed blobs based on block ID | Block blobs are mutable.

Downloading and Copying blobs: GetBlob | Range reads are supported | CopyBlob

Listing and Deleting Blobs: ListBlobs | DeleteBlob (Drop containers with several objects using a single atomic call)

Client Libraries offer many features.

Versioning: Snapshots | Can be read, copied/restored or deleted.

Concurrency: Optimistic concurrency - Timestamps/ETags | Pessimistic concurrency - Leases | Last Writer Wins (Not recommended). There is a white paper available on this subject.

User Experience

The Field Museum | Nasuni’s customer | Azure solution - GRS implemented, Data volume managed by a combination of Nasuni File Service and Azure Storage. Requirement: File or Block access | Global Repository | Accessible Anywhere | Edge Caching | Global File Locking | Enhanced Security. This is a partner integrated solution.

Data movement: AzCopy tool simplifies data migration at scale | Data movement library supports developers.

Delegated Authorisation: SAS Tokens enable direct access by client apps; recommend for application data access instead of storage account key.

Encryption at Rest: Blob storage service encryption | Application encryption.

Full Text Search: Azure Search - Full text search for blobs (and tables) | Index data and metadata.

Cloud Tiered Storage: Two Tiers: Hot and Cool | Pricing to match your workload

User Experience

Seminole | Altavault ecosystem for Azure - Archiving, kind of similar to StorSimple kind solution. Tiered storage in a hybrid context. Quick Onboarding | Retire Tape Drive Solution | Faster Backup and Restores.

Backup and Recovery solution: Backup | Target | DR | Migration

### **Build microservices and container solutions using Azure Service Fabric and Azure Container Service | Tejas Karmarkar**

Cloud considerations: Don’t own hardware | Build to fail | Unpredictable scale | Managing services is harder than building them

Microservices approach: Continually evolve apps | Faster delivery of fixes

Scaling by replication is painful in traditional application approach; Microservices can help.

Traditional design: Upgrades can be cumbersome, hard to scale data access, expensive to scale application. One fine example is MS Office vs Office365

Microservices: Application composed of services. Stateless presentation services, other services with stageful data services.

MSA benefits: Scale independently | Independent Deployments vs Different Technology Stacks | Conflicting updates

Build —> MSA —> Deploy

SQL DB | DocDB, etc. are built using Microservices architecture.

Azure Service Fabric explorer can provide peek into how nodes (VMs) are deployed across FD and UD

Packages automatically moved from a failed node to another active node on the fly, after recording the failure with an error message. App cleans up itself inside the Azure Service Fabric cluster, if stopped.

ASF: Supports 1000s of nodes | Is self repairing | Scales in and out. BYOA is possible within ASF, but some control could be lost wherein the ASF cannot have the necessary insight into the code to report state or bug.

Categories of services in ASF: Guest executables and guest containers | Stateless services | Stageful services

ASF frameworks: Reliable services | Reliable actors | .NET Core | Any code

Note: Code governance in terms of regulating updates to client SDKs, APIs, other program modules, etc., is necessary for stateful applications. Although application services can be transferred, it causes some level of partial downtime in terms of latency. Auto updates to the underlying framework and libraries can many times be counterproductive.

A single service fabric application design is possible for stateless and state-ful applications, by determining the data repo.

ASF does not peek inside the app code and inform. It only ensures updates happen to only one UD at a time and verified prior to updating other UDs. While FD pairing is on binary, UD pairing is can be larger. ASF ensures CI-CD to push the code simultaneously when the system is being upgraded across the ASF cluster nodes. Upgrades could be seamless.

Ref: ASF introduction video on [aure.com](http://aure.com/). ASF can support Azure, On-Premise, and other cloud platforms.

### **Build modern cloud based web and mobile applications on Azure App Service | Nazim Lala**

Azure App Service: Web Apps | Mobile Apps | Functions | API Apps

Features and Capabilities: Enterprise grade | Fully managed | Built for DevOps

Customer use cases

[Jet.com](http://jet.com/) | Moving from code to production in minutes

DevOps workflow IS NOT Dev workflow in terms of the functions and landscapes involved. For example, the development process of Dev-QAS-Pre-Prod-UAT-Prod is not the same as the DevOps workflow, which essentially focusses on CI-CD and some kind of a tandem process between build-staging-production. Testing and Quality Assurance are not essentially main aspects of DevOps and have more to do with Devs rather than Ops.

App Service Plan is the basic unit of compute for App service.

A Webhook in GitHub is kind of an SCM.

Demo | Site Slots: Allows the deploy code to staging.

Slot setting helps bind configuration changes to the code in the slot regardless of the code swap between the concerned slot and another slot. An example being a slot setting ensures sticky configuration carried with the staged code being moved to production.

Staging slots’ impact on application routing, and TLS termination?

AAD: On-Premise accounts can be replicated to and federated by AAD for applications or services that are hosted across either environments. TP authentication providers can also be integrated with AAD relaying application access requests to them and accept the resulting access tokens, for accessing the hosted application on Azure, On-Premise or on some other cloud platform.

MFA is defaulted whenever TP authentication is invoked for application access. JWT can be passed for authenticating and restricting specific user groups.

Demo: App Service Authentication and Authorization

App Service on Linux in preview

User voice forums has features list with votes based on which Azure services are developed and enhanced.

Demo: App Service on Linux | Web App on Linux

Container creation is defaulted; based on Docker container engine; App is hosted on a Docker container at the backend.

It is no more code building with CI and Git. It is building a Docker image from a Git.

SCM does source control management, besides app monitoring, usually associated to <appdomainname>.[scm.azurewebsites.net](http://scm.azurewebsites.net/)

You can SSH into SCM code console. SCM codes are in a separate container, in a persistent storage.

Docker container configuration is exclusive to App Service on Linux. Built-in, Docker Hub, Private Registry are the image source choices available, which helps with choosing codes from languages not supported in default choice of Built-in container engine.

Docker client is available for Mac and Linux.

[GitHub.com/azure-app-service](http://github.com/azure-app-service) has all the blessed or built-in images for multi-platform applications.

Docker images are built up of layers, hence incremental pushes will not consume more time.

Once there is a code change and push, it will be served from a new container. All requests from then on will be directed to the new container. The old container shall still be serving existing service requests until it drains after which it will be pulled over. This is possible because of Dockerized images being built and deployed instead of direct code.

ASP value proposition: Web + Mobile + Functions + API

Secure Apps and Secure NW Access

Hybrid across On-Premises and Cloud. Private access can be setup for specific apps and public for others, leveraging the connection and integration between the On-Prem and Azure.

Scale up and scale out around the globe.

aka.ms/tryappservice - free try

## Day 2

### **Labs on demand | Available for 29 more days, accessible from anywhere, available for all the approved registration accounts, at** [**https://bangalore.techsummit.microsoft.com/Pages/Index/LabsOnDemand**](https://bangalore.techsummit.microsoft.com/Pages/Index/LabsOnDemand)

### **Port your AWS knowledge to Azure | Colin Marshall,** [**v-colin@opsgility.com**](mailto:v-colin@opsgility.com)

Future belongs to a multi-cloud based solution environment coupled with On-Premise and/or hosted deployments in TPDC/Colo, in a true hybrid, heterogenous, consumer/end user and business data-driven delivery model

Azure strengths: Enterprise Agreements | Hybrid connectivity

AWS wants customers to run everything from the cloud

Not everything that runs on IaaS runs on PasS

4 pillars: Application innovation | Data and intelligence | Openness and flexibility | Trust

Biggest expense in the cloud is the compute (60-80% of the budget)?

Build on PaaS: Existing frameworks | Mircoservices | Web and Mobile | Serverless compute

Build on IaaS: Storage | VMs | Networking

Azure IoT suite, Bot framework, Cognitive services, Azure marketplace (for specific services outside of Azure umbrella or apps and tools for specific purpose to an organization-enterprise)

Azure Functions = AWS Lambda

Build and run open source solutions: Any tool, application, and framework

Azure regions: 38 regions worldwide, 34 online

Government regulations on regions and their pairs: Singapore Government wanted a workaround performed by Microsoft because they (Southeast Asia) have their partner DC in East Asia (Hong Kong) which is under China’s governance.

Dark fiber: Lying around to be used when needed.

AWS Regions: 18 worldwide, 16 online

One of the differences is in storage. Azure offers storage accounts with LRS, GRS replication.

AWS regions are smaller owned and Colo DCs called AZs - 47 in total. Azure, on the other hand owns complete, larger DCs.

AAD: SSO to 2800+ applications, has RBAC enabled access.

JSON is understood by AWS and Azure.

AWS Resource Groups are different from Azure Resource Groups. Tags (Metadata) is the same.

AWS offers a single Gov region/DC service. Azure offers at least 5 - 6 DCs in couple of regions.

Azure bills by every minute but AWS bills by every hour.

If data sovereignty is not an issue, deploying to different regions helps save cost, due to price variations, a long term savings.

Pricing calculators: [azure.microsoft.com/en-us/pricing/calculator](http://azure.microsoft.com/en-us/pricing/calculator) [calculator.s3.amazonaws.com/index.html](http://calculator.s3.amazonaws.com/index.html). They both different deals in the background, especially for customers with Enterprise Agreement or long term hosting commitment.

AAD vs AWS DS: Azure strengths - More objects support | Multi-tenant PaaS solution leveraging IDM.

Modern identity with AAD: 2800+ external authorisation providers available alongside AAD, to be leveraged by enterprises.

Deployment choice in Azure: CLI, REST API, PS, ARM Templates, VS, besides the portal

Networking at a glance: VNet/VPC, CDN/CloudFront, DirectConnect/ExpressRoute (Lower latency|More privacy), Route53/TrafficManager-DNS

VNet NSGs (VNet, Subnet, VM) / VPC Security Groups (Subnet) and ACLs (VM) | Hybrid connectiivity with IPSec VPN, PPTP, Peering, and ExpressRoute / Hybrid connectivity with VPN/VPN peering, and Direct Connect | BYO DNS at VNet or VM / BYO DNS, NetBIOS, NTP with DHCP option sets.

NSGs: Acts as an internal and external firewall, from VM to the network core including the CE gateway.

DNAT/RNAT services help reduce number of Public Its used/exposed, reducing the vulnerability footprint and save some cost. Remember the cost of managing a NAT service/device/appliance.

35 ExpressRoute locations worldwide, nearly doubled peering locations and partners than any other CSPs.

VM/EC2 | ACS/EC2 CS | Web Apps/Elastic Beanstalk, ASR, Azure Backup/No AWS counterpart except for the AWS Server Migration Services (recently added and moved over to DMS), Marketplace/Lightsall, Functions,Web Jobs, Logic Apps/Lambda.

Public IP/Elastic IP, ALB/ELB

AWS image sizes/instance sizes are specific to the operation/function to be run. Azure is not specific but provides high-level guidelines around instance selection in terms of T-Shirt sizes and Azure Compute Units.

S3 natively supports static website/technically possible in Blob but no default document, EFS used NFS (Linux only), Azure File Storage uses SMB 3.0

Azure deploys in HDD by default. AWS deploys in SSD by default. Azure Premium storage is a real use case for Scale Up architecture where it can pair with On-Premise servers when they lack some resource/capability.

Microsoft tops the chart of the organisations that contributed the most to the open source community, followed by FB and Docker, as recent as Sep 2016.

vLAN are VNet equivalents on-premise. This indicates that there could be multiple subnets inside a vLAN.

AzCLI available from within VS Code.

Each Azure subscription is a separate tenant, from boundary of services isolation. Nevertheless, there is a boundary of billing isolation encompassing multiple subscriptions inside what is called an Azure Account, especially and Enterprise Account.

Demo: VNet create | Public IP create | LB create | LB inbound/outbound/probe/rule | NSG create, rule create | NIC create | Associate Set, NSG ID with the VM | Configure AVset

Azure File Service based mount points shall be used to connect to Azure Storage Accounts, which can then be associated with a Linux VM. Apache shall help build and deploy a simple web page. Test the load balancing works, by reopening the page.

Azure Resource Group project inside VS provides bunch of pre-built templates for deployment of resources onto Azure. These are JSON script codes, which will then be run from within VS to be hosted on a specific Azure subscription, without logging onto the portal.

### **Microsoft Azure networking:Getting things connected | Narayan Annamalai**

38 Azure Regions | 2.5 Mill KM of Fiber | 34 Express Route locations

SDN: Stripping intelligence off NW hardware. NVF is leveraged for purposes such as DNS, load balancing, routing, commodiitized hardware (No more HW OEM)

ToR —> Spines —> DCR —> internet —> Strategically locate Edge site  —> end users

ToR —> Spines —> DCR —> MPLS WAN cloud type circuits with strong PoPs around the globe —> Other Azure DCs

Microsoft has laid under the sea cables and optic fibre for intercontinental connectivity.

aka.ms/cloud-disruption

VDC: Cloud becomes another location, like another DC confirming that cloud can act as.a conduit and exist within this enterprise eco-system involving partner services, vendors, and customers forming it a truly hybrid, global solution.

VNet is basically a trust boundary. Mix them with any of the IaaS services; extension to and leveraged by PaaS services is coming soon.

Small organisations mostly live in cloud, but medium and large organisations/enteprise live on-premise and on cloud, hence the need for a hybrid connectivity for infra extension or collaboration among independent modules/services in either sides of the hybrid network. Hybrid solution is made possible by leveraging VPN/ExpressRoute services

Custom routes (UDR), Custom traffic shaping with Network Virtual Appliances?

VNet to VNet through peering/gateway. Peering employs high-speed switching devices for low-latency high-speed requirements. On the other hand, Gateways really help build a true DR replica of a given solution in a different site/DC in a different region. Hub and Spoke technique can also be leveraged?

VMs performance increased by improving the VNets’ performance.

SRIOV lets use of the hardware more efficiently. FPGAs on these Hyper-V server hardware NICs are leveraged for high-speed packet switching and other advanced and powerful features between VMs within the same hosts. 30-90 ms of latency per packet (1KB?) can be achieved using this technique.

An equivalent to SDN could be the Cisco ASA appliance series, which control the abilities and advanced controls within the hardware to be unlocked and leveraged seamlessly.

Azure DNS: all common DNS record types | Flexible DNS management. Microsoft is not the registrar but host for the customer DNS system itself.

IPv6: Native support to VMs | Maximize the reach of your Azure applications. O365 share point online editing of documents is served by a IPv6 endpoint, built and managed by Azure.

Benefits: IPv6 clients can directly connect with our solution; same with IoT devices as well. Other v6 devices on the cloud can connect with your v4 services using SNAT feature available on Azure.

ALB: Different algorithms based; ATM is but ALB done at geographical scale. ATM works at DNS level (Which OSI Layer?). Besides, there is Azure Application Gateway for Layer 7 load balancing that could be leveraged by suitable applications. Hosting TPAppliances on Azure is also absolutely fine.

Application Firewall services can also be leveraged: in preview currently.

Multiple VIPs for LB: VIP binding per application group (LB Set) is possible.

NIC enhancements: MAC persistence (For licensed applications) | Updates (NICs addition/removal, Single and Multi-NIC VMs in a single Availability Set) | LB for all NICs | Multiple IPs on a NIC. NICs in a multi-NIC VM set, NICs can be grouped and defined to be used as external and internal load balancing channels.

Aviatrix and Microsoft solved NW latency issues of Hyatt group of hotels by leveraging Microsoft network presence and Aviatrix’s enterprise cloud management suite? This helped in improving application performance used by internal services management teams and their customers.

Cross premised connectivity scenarios: Scenario 1 - Office connected to Azure cloud infrastructure using VPN tunnels, while Azure PaaS and SaaS services portfolio can be leveraged/accessed using ExpressRoute which will be passing via the organisation/enteprise private WAN. This makes it possible for the OPEN and CLOSED business worlds to live together within the same organisation. Active - Active tunnels across different VPN circuits is possible. Chennai and Mumbai DCs are enabled by Res presently.

Sovereign nations restrict hosting of Azure services on Azure DCs. They host on behalf of Microsoft, with strict network boundaries. China and Beijing are an example. Locally registered cloud providers will be Microsoft’s cloud conduits in these sovereign nations.

Monitoring and Diagnostics: Azure Network watcher - can manage VNets, ERs, and Application Gateways. Metrics | Logs | Diagnostics | Alert | Topology. Insights into routing paths/hops, log edits, allows/denials, etc. Variable packet capture, NSG Flow logs, IP Flow verify, VPN dogs, Network subscription lists, Single pane to configure logs and metrics.

### **Configuring your heterogeneous environment with PowerShell and DSC | Ritesh Modi | Senior Technology Evangelist | @automationnext**

Candidates: Windows | Linux | Mac

W2k16 - Built-in layers of security | SDDC | Cloud-ready application platform

Windows NT - No PowerShell, Only DOS .bat/VB Scripts | W2k; 2k3 - PowerShell launched with limited capabilities |W2k8,W2k12 - PowerShell based full-automation | W2k16 - PowerShell based cloud-automation

DevOps tooling possible with PowerShell DSC

PowerShell can help manage Win/Lin

PowerShell acts as a glue to bind to the various services, files, configuration containers such as registry etc., by leveraging different APIs.

PSObject abstracts/ front ends requests by extending them to the actual services such as COM, WMI, .NET, XML, CSV, JSON, etc. PSObjects are discoverable

Invoke-webrequest command-let provides us with a web-object. Content of this object is in JSON like name-value pair format but not exactly JSON code. Gettype method can help validate. Get-member is alaternative method for lookup of object types.

Coding: Abstract coding (based on a scripted environment such as PowerShell, BASH, CSH, etc.,) | Actual coding / system level programming (based on a natural language such as C# or Java)

.NET Core - small subset of .NET framework usable for coding and scripting by leveraging PowerShell for Linux based on .NET core.

PowerShell on linux can be run from an SSH console (Putty). Same strings and methods can be fetches on Linux PS as in Win PS.

Now, converting the string into JSON - convertfrom-json can help. After conversion a lot of metadata could be retrieved as objects (3D).

JSON can be retrieved by using REST method also. This is by calling the specific REST API within the invoke-webrequest command let.

PS is open sourced to communities for inclusivity in terms of enhancements, bug fixes, etc.

PS is portable and works the same on all Linux distros despite having different versions for the various flavours. Examples being $iswindows, $islinux, $PSVersionTable results.

Linux commands can be run within PS console. However, an equivalent exists in PS called $pwd. ls|get-childitem is another example.

Some of the commands are common across the environments, but some other are unique to each of the environments.

get-command, get-help work across. get-command verb “<*VERB*>” will list the specific commandments starting with the string “verb”.

-listavailable parameter lists the portfolio, active, inactive, and suspended services.

item/child-item/content helps working with file system hierarchies in Linux.

sudo rm/remove-item are equivalents.

PS install on Linux: sudo su, curl PS GitHub URI, sudo YUM install -y powershell | sudo apt-get install <libraries for ps>, sudo dpkg <ps package>

Cloud era - management landscape: Power | Velocity | Ubiquity

PS 5.0 introduced lot of improvements to the DSC and JEA (Just Enough Administration) modules.

PS 5.1 most enhancements to DSC. PS 6.0, not yet in GA is the most current one.

Power: DSC | PS Classes | New debugging/editing features | DSC resources

import-dscresource -modulename -moduleversion

New-cimsessionoption -usessl

New-cimsession

Start-dscconfiguration -cimsession -oath -wait -verbose

get-dsclocalconfigurationmanager updates the configuration refresh frequency

PS DSC Modes: Push | Pull

DSC configuration can be pushed from a Win system to Lin systems and vice versa. An example being to push a file onto a specific Linux distro FS.

new-object command let

DSC internally uses WSMAN for Linux remoting.

PSClasses:

works on all environments irrespective of whether it runs over a coding/scripting platform.

enum color | class car, ToString | create a function new-car, invoke it as a command-let new-car

PSGallery helps find modules and allows new module inclusion, is community driven

Leverage WMFs to install latest PS versions on older Windows systems such as W2k8

Pester is a community tool for testing PS scripts, now included with W2k16. Simple test cases can be done with number math such as addition or subtraction.

Pester is available out of the box in all environments. DSC has a reporting mechanism as well.

VSCode is the cross-platform tool for PowerShell scripting for heterogeneous environments.

ARM Templates and PowerShell DSC

1. Store DSC related software packages/files/configurations that need to be sent to clients, in an Azure Storage Account.
2. Configure Get-DscLocalConfigurationManager to point to the Azure storage account from where to pull the packages/files/configurations from, during  a deployment
3. Call PowerShell DSC as a resource from within the ARM template. Actually, DSC is a child resource under VM resource type.