```
"Variables": {
      "Session": "Infrastructure as Code",
      "Speaker": "Marcel Zehner",
      "Details": [
            {"Company": "itnetX"},
            {"Job-Title": "CTO"},
            {"Microsoft-Most-Valuable-Professional": 1},
            {"Microsoft-Regional-Director": 1},
            {"Twitter": "@marcelzehner"},
            {"Blog": "marcelzehner.ch"},
      ]
}
```

Some of today's IT challenges

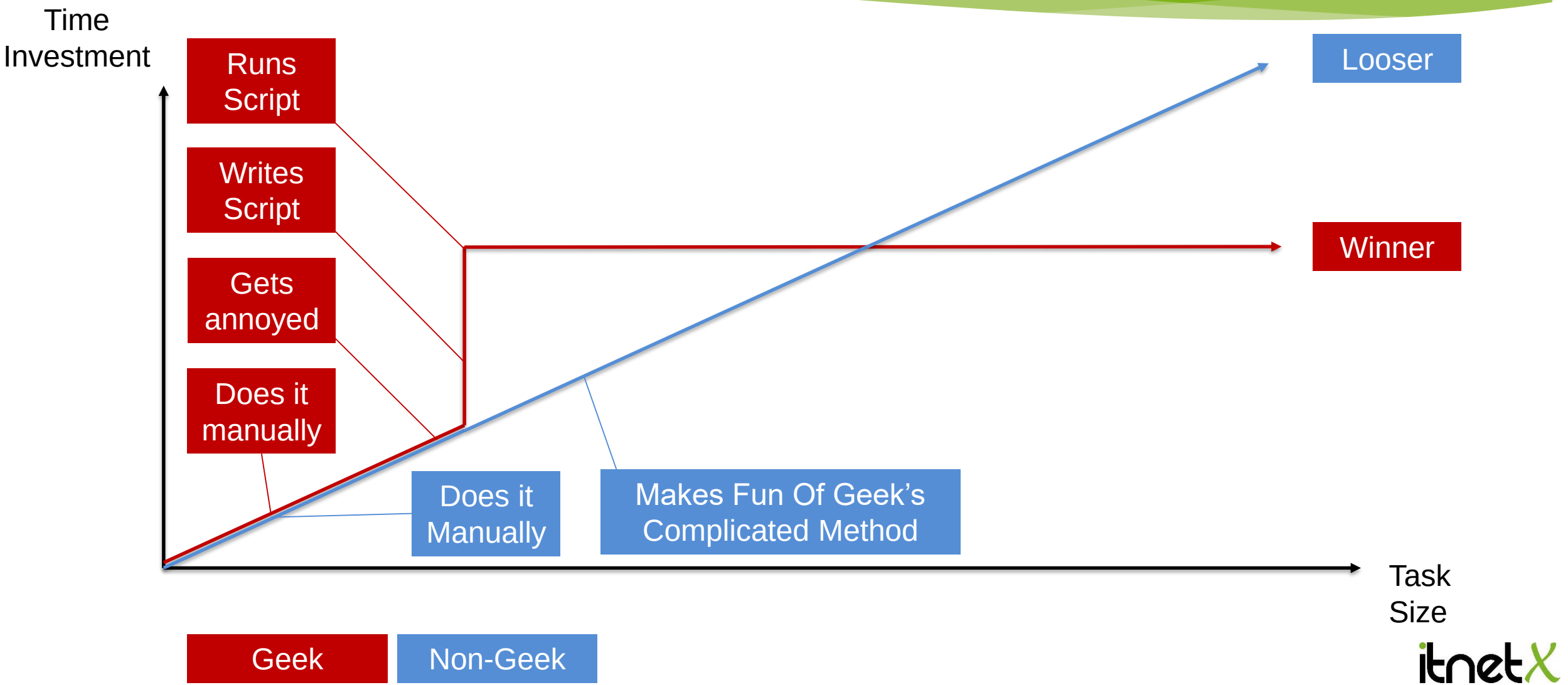
- More and more complex infrastructures
- Manual configuration
  - Slow, errors, inconsistent and boring
  - Humans have too many permissions
- Configuration drifts
  - Admins don't follow the processes
  - No change management/tracking
- Less people, budget and time
- And tons of other challenges …

itnetX

Automation is your friend!

- Automation solves some of the challenges
  - Let machines do the boring work for you
  - Invest time in innovation!
- Automate it processes that you use more than once
  - Standardize, then automate

“Automated vs. Manual” or “Geek vs. Non-Geek”



Infrastructure

- What about the infrastructure?
  - Should be provisioned in an automated way as well
- 2 IaC approaches
  - Imperative (procedural)
    - Configuration is done step-by-step
    - Automation scripts, workflows or runbooks
  - Declarative (functional)
    - Final state of system is described
    - Automation process configures everything as described

2 Approaches for IaC

imperative

```
New-AzureVM –VM $myVM
New-AzureStorageAccount –StorageAccountName $acct
Set-AzureVNetConfig –ConfigurationPath -Path
```

declarative

```
{
 "$schema": "https://../deploymentTemplate.json#",
 "contentVersion": "1.0.0.0",
 "parameters": {},
 "variables": {},
 "resources": [],
 "outputs": {}
}
```

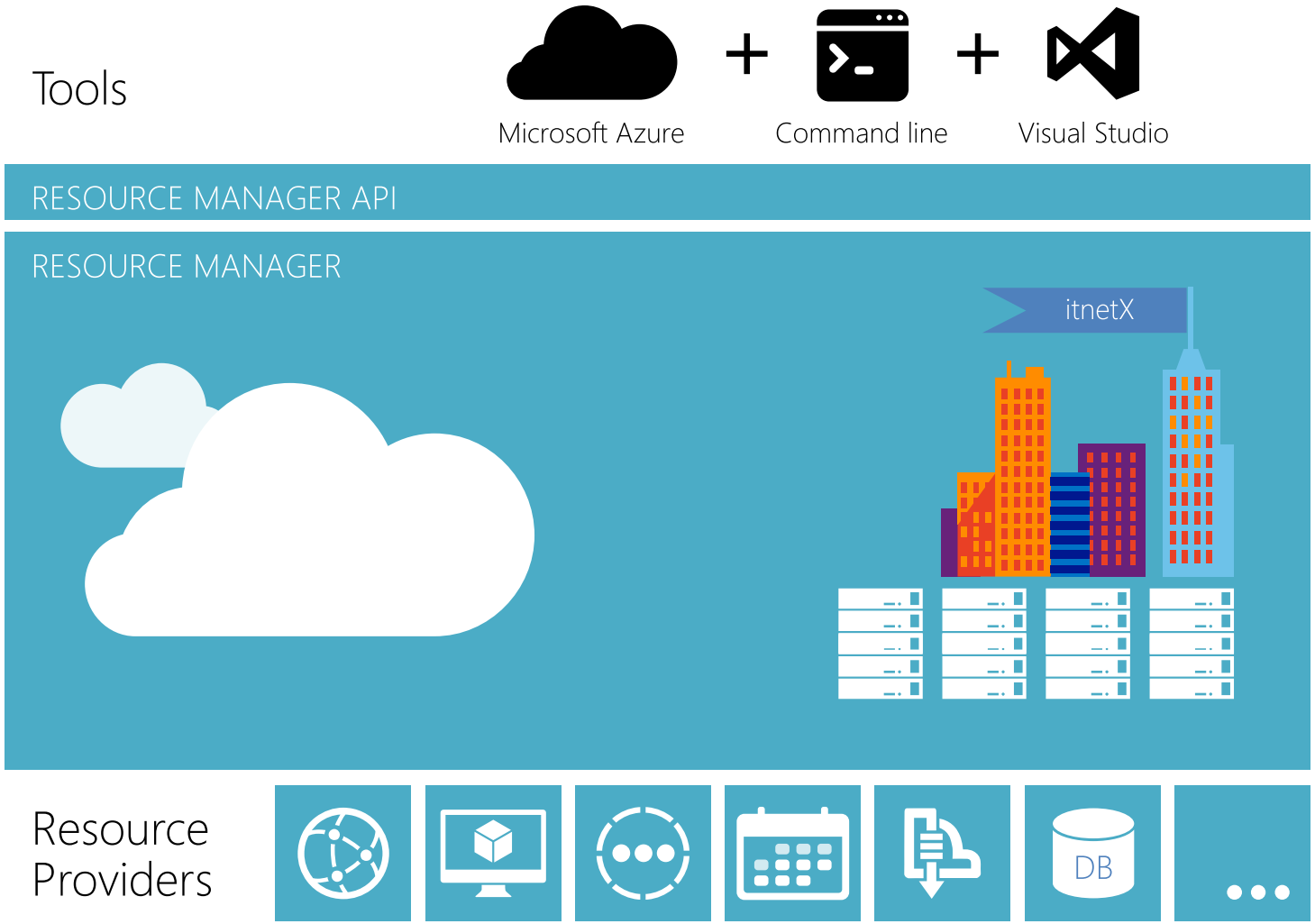
itnetX

Azure Resource Manager

Azure Resource Manager

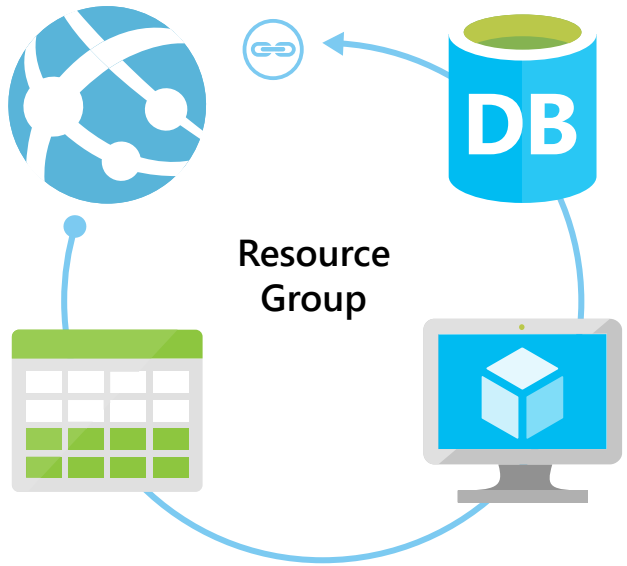
- ARM manages resource providers
- Resource providers manage Azure resources
  - Azure virtual network
  - Azure SQL database
  - Azure LogicApp
  - Azure virtual machine
  - Etc.

Azure Resource Manager



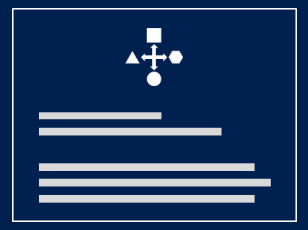
Azure Resource Manager

- Resources can be grouped in resource groups
  - Containers
  - Resources that share the same lifecycle
  - Every resource exists in only 1 resource group
  - Access delegation to resource group possible
    - RBAC



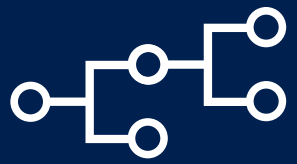
itnetX

One Azure Ecosystem



Applications

- Windows | Linux
- Java, PHP, .NET, …
- IaaS
- PaaS
- Containers



Operations

- Templates
- PowerShell, CLI
- Puppet, Chef, DSC
- Metrics
- Diagnostics

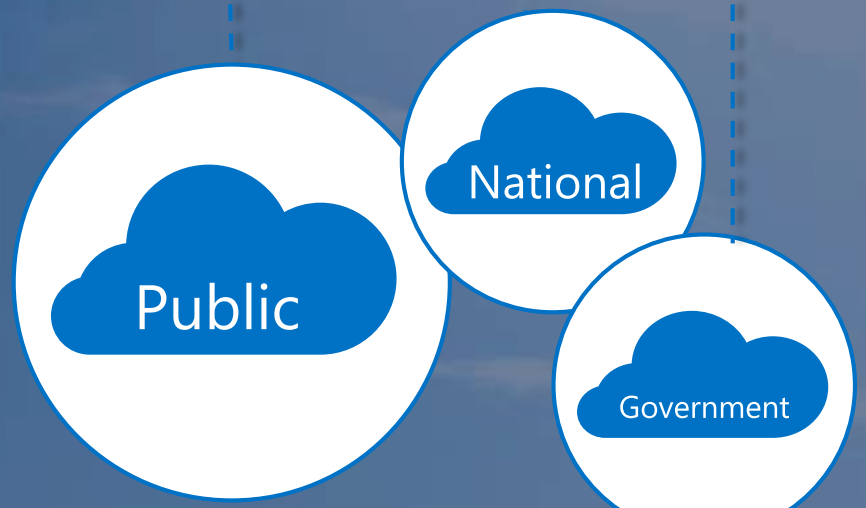
Tools

- Visual Studio
- Eclipse
- …

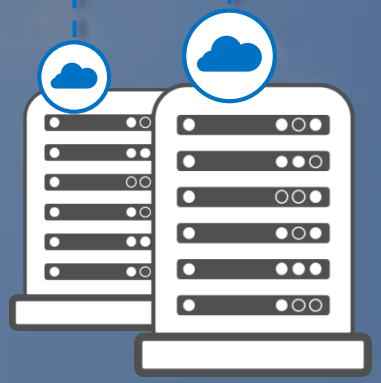
Experiences

- Portal
- Gallery
- RBAC
- GitHub
- …

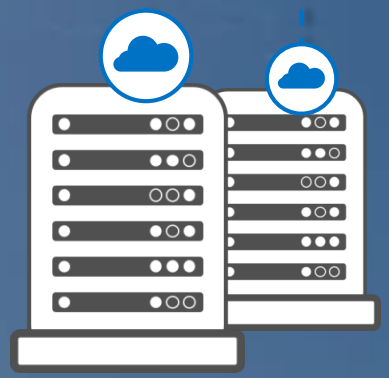
Azure Ecosystem



Microsoft Azure



Service Providers



Enterprises

Microsoft Azure Stack

ARM Templates

- ARM APIs understand human-readable JSON Files
  - Declarative
- Can be created in a variety of tools
  - Visual Studio, Visual Studio Code, Notepad etc.
- Lifecycle
  - Describe the app
  - Test the deployment in test environment inside out
  - Deploy the exact same configuration to the production environment

Demo

Azure Resource Manager

itnetX

Powershell DSC

- Declarative approach for VM configurations
  - VM can be deployed with ARM
  - Afterwards the VM needs some configurations
- MOF file assigned to VM
  - Windows and Linux
  - Contains configurations based on DSC resources
  - Can be pushed or pulled
  - Local configuration manager consumes the configuration
    - Monitor only, Remediate

- Archive Resource
- Environment Resource
- File Resource
- Group Resource
- Log Resource
- Package Resource
- Registry Resource
- Script Resource
- Service Resource
- User Resource
- WindowsFeature Resource
- WindowsProcess Resource

itnetX

Powershell DSC with Azure Automation

- Create DSC Configuration File
  - Import into Azure Automation
- Compile configuration file into node configurations
  - One per role
- Onboard DSC nodes to Azure Automation pull server
  - Azure VMs (VM DSC extension)
  - Any other VMs (on-prem)
- Assign node configuration to VMs

itnetX

Powershell DSC Step-by-Step

Configurations

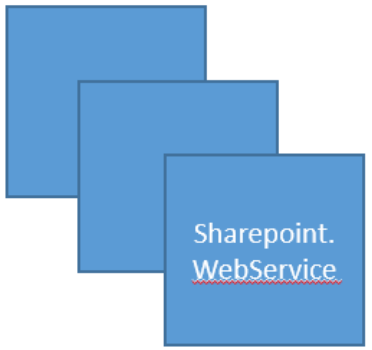
```
Configuration SharePoint {

  Node WebService {

    #Install the IIS Role
    WindowsFeature IIS {
      Ensure = "Present"
      Name = "Web-Server"
    }

    #Install ASP.NET 4.5
    WindowsFeature ASP {
      Ensure = "Present"
      Name = "Web-Asp-Net45"
    }

  }
}
```

1 or more per automation account

Node Configurations
(.MOF configuration documents)

Compiled, put on pull server

(via compilation jobs)



1 or more per Configuration

Applied

(via node pulls)

Nodes

1 or more per Node Configuration

itnetX

Demo

Powershell DSC with Azure Automation

itnetX

Conclusion

- Use declarative approach where possible – think modern!
  - Azure Resource Manager (ARM)
  - Powershell DSC
- Use source control and versioning for your configuration files
  - VSO, TFS, GitHub etc.
- Visual Studio and Powershell are your friends, not your enemies!
  - Even if you are an ITPro!

```
"Variables": {
      "Session": "Infrastructure as Code",
      "Speaker": "Marcel Zehner",
      "Details": [
            {"Company": "itnetX"},
            {"Job-Title": "CTO"},
            {"Microsoft-Most-Valuable-Professional": 1},
            {"Microsoft-Regional-Director": 1},
            {"Twitter": "@marcelzehner"},
            {"Blog": "marcelzehner.ch"},
      ]
}
```