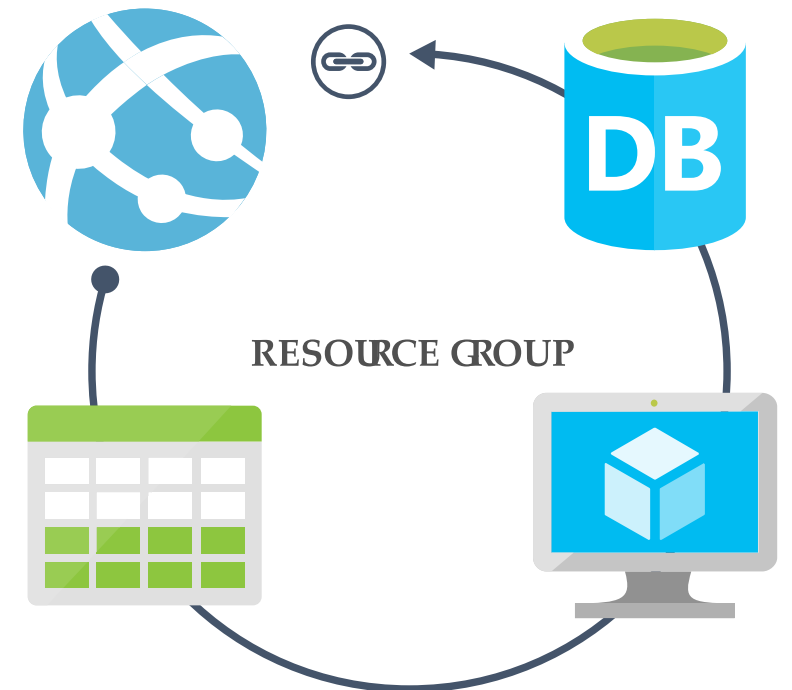


Infrastructure as Code:

Azure Resource Manager inside out

Henry Been



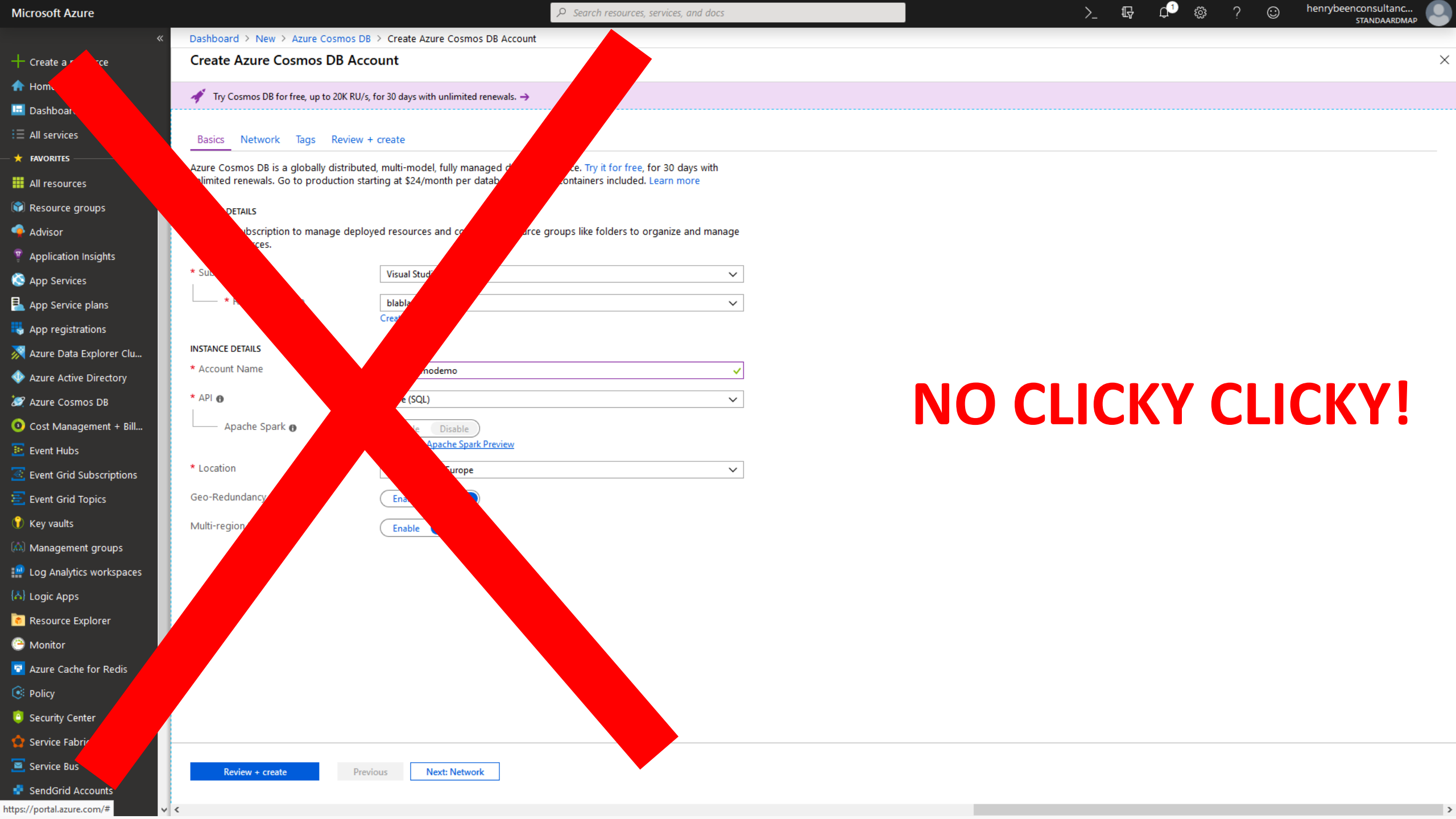
WONDERING **WHO** IS THAT GUY?



HENRY BEEN

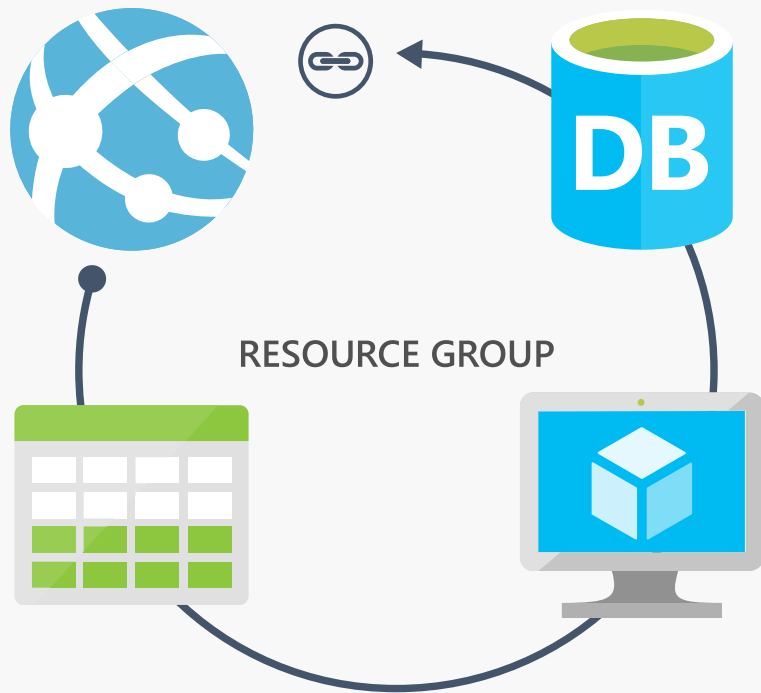
Independent Devops & Azure Architect
Microsoft MVP & MCT

E: henry@azurespecialist.nl
T: [@henry_been](#)
L: [linkedin.com/in/henrybeen](https://www.linkedin.com/in/henrybeen)
W: henrybeen.nl



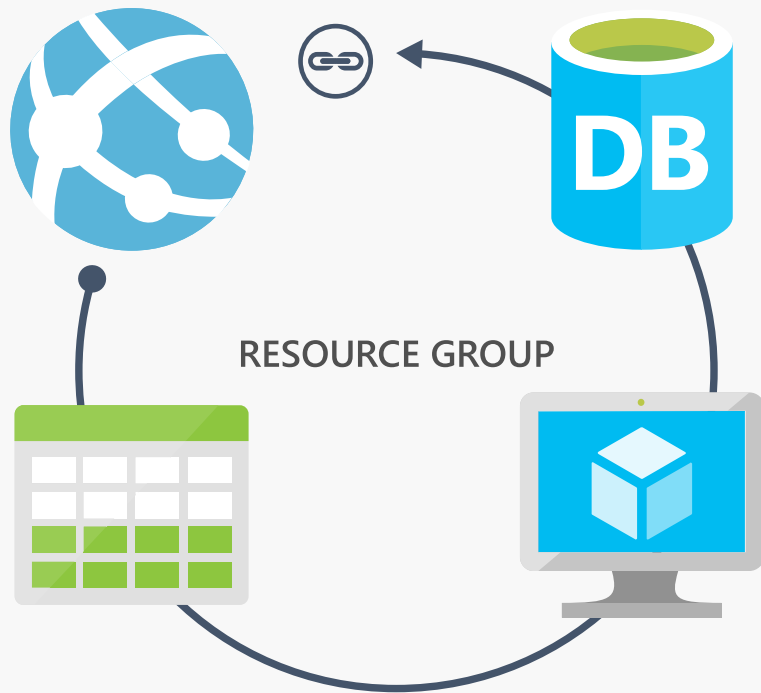
NO CLICKY CLICKY!

Azure Resource Manager (ARM)



- Unified management API for all Azure resources
- Templatize application deployment and configuration

Azure Resource Manager (ARM)



- Resources & Resource Providers
- Infrastructure as Code
- Automating deployments
- Policy

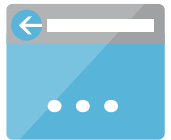
Everything is a resource



App Services



Virtual Machines



IP Addresses



Dashboards



Alert Rules

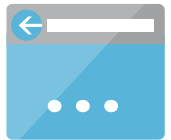
Everything is a resource



App Services



Virtual Machines



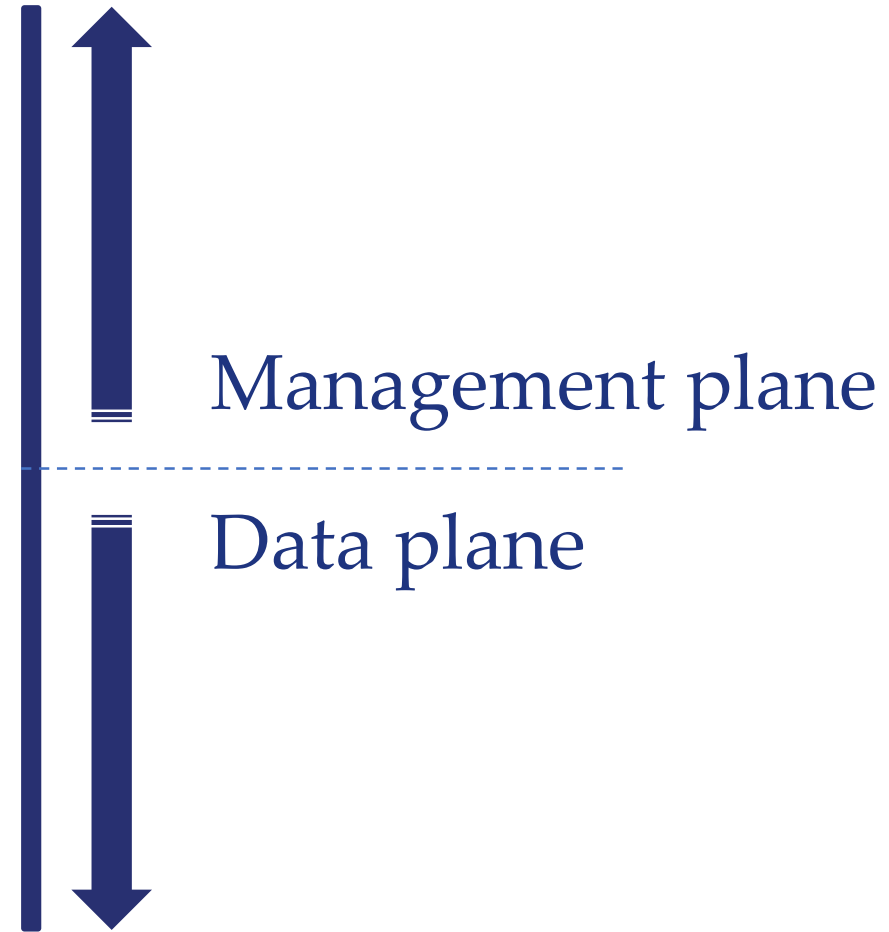
IP Addresses



Dashboards



Alert Rules



Everything is a resource



App Services



Virtual Machines



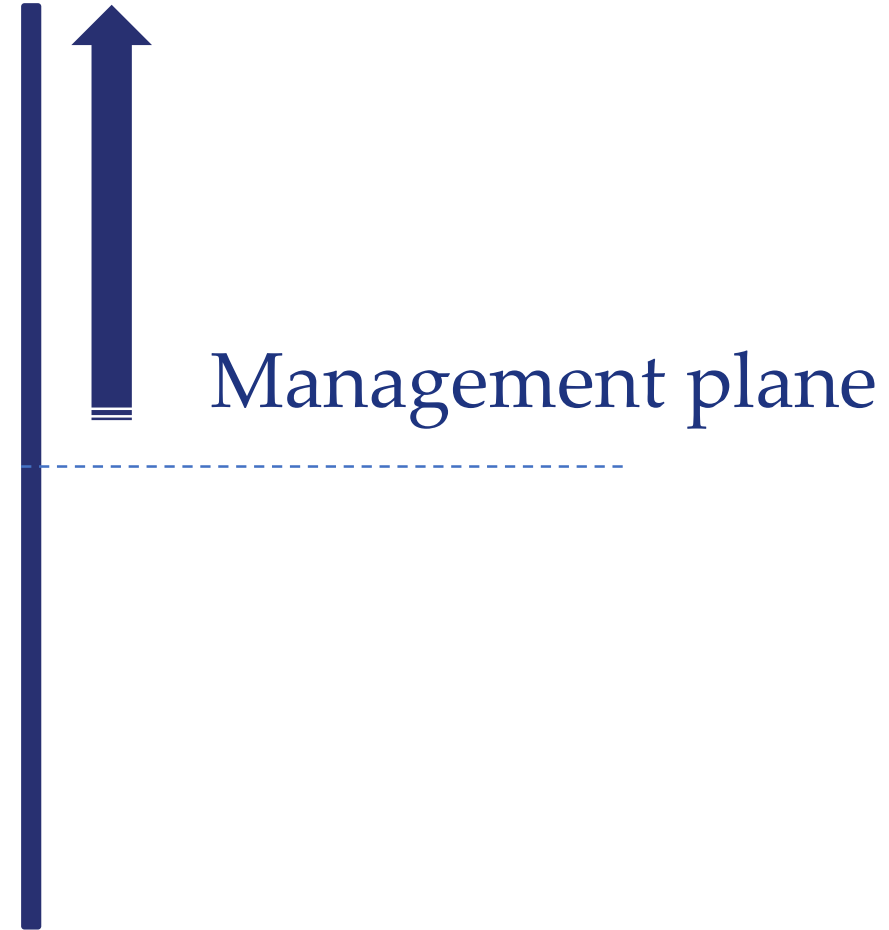
IP Addresses



Dashboards



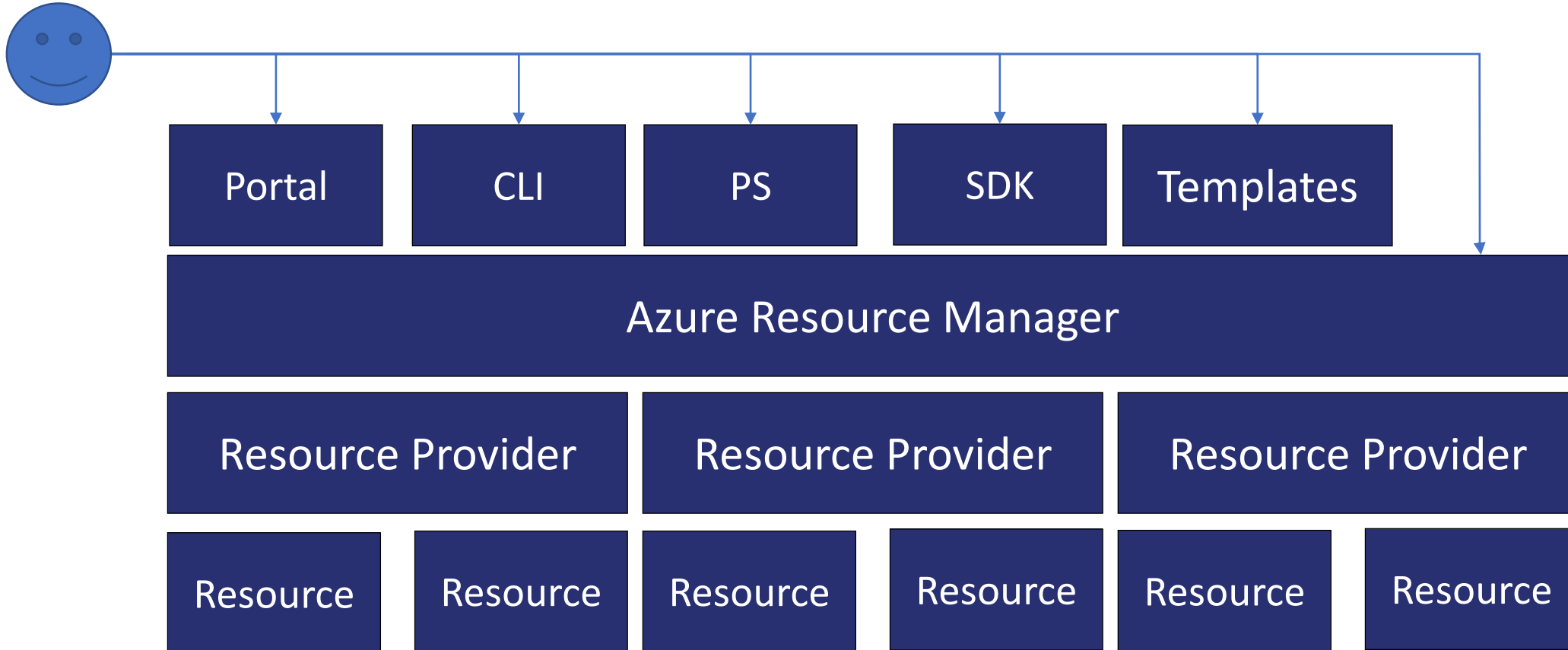
Alert Rules



Resource groupings



Resource providers



Addressing a resource

https://management.azure.com
/subscriptions/{subscriptionId}
/resourceGroups/{resourceGroupName}
/providers/{providerName}
/{resourceType}/{resourceName}
?api-version={apiVersion}

Addressing a resource

https://management.azure.com
/subscriptions/{subscriptionId}
/resourceGroups/{resourceGroupName}
/providers/{providerName}
/{resourceType}/{resourceName}
/{subResourceType}/{subResourceName}
?api-version={apiVersion}

PUT <https://management.azure.com/subscriptions/guid/resourceGroups/example/providers/Microsoft.ServiceBus/namespaces/mySpace?api-version=2017-04-01>

```
{
  "sku": {
    "name": "Standard",
    "tier": "Standard"
  },
  "location": "West Europe",
  "tags": {
    "myTag": "TagsCanBeUseful"
  }
}
```

```
{
  "sku": {
    "name": "Standard",
    "tier": "Standard"
  },
  "id": "/subscriptions/guid/resourceGroups/example/providers/Microsoft.ServiceBus/....",
  "name": "mySpace",
  "type": "Microsoft.ServiceBus/Namespaces",
  "location": "West Europe",
  "tags": { .. },
  "properties": {
    "provisioningState": "Created",
    "metricId": "5f750a97-50d9-4e36-8081-c9ee4c0210d4:mySpace",
    "createdAt": "2019-06-07T22:26:36.76Z",
    "updatedAt": "2019-06-07T22:26:36.76Z",
    "serviceBusEndpoint": "https://mySpace.servicebus.windows-int.net:443/"
  }
}
```



Good thing there
is a **reference!**

<https://docs.microsoft.com/en-us/azure/templates/>

More convenient ways...

```
Connect-AzureAzAccount
```

```
New-AzResourceGroup -Name myGroup -Location "West Europe"
```

```
New-AzVm `
```

```
-ResourceGroupName "myGroup" `
```

```
-Name "myVM" `
```

```
-Location "West Europe" `
```

```
-VirtualNetworkName "myVnet" `
```

```
-SubnetName "mySubnet" `
```

```
-SecurityGroupName "myNetworkSecurityGroup" `
```

```
-PublicIpAddressName "myPublicIpAddress" `
```

```
-OpenPorts 80,3389
```


More convenient ways...

```
az login
```

```
az group create --name myGroup --location westeurope
```

```
az vm create \  
  --resource-group myGroup \  
  --name myVM \  
  --image win2016datacenter \  
  --admin-username azureuser \  
  --admin-password myPassword12
```

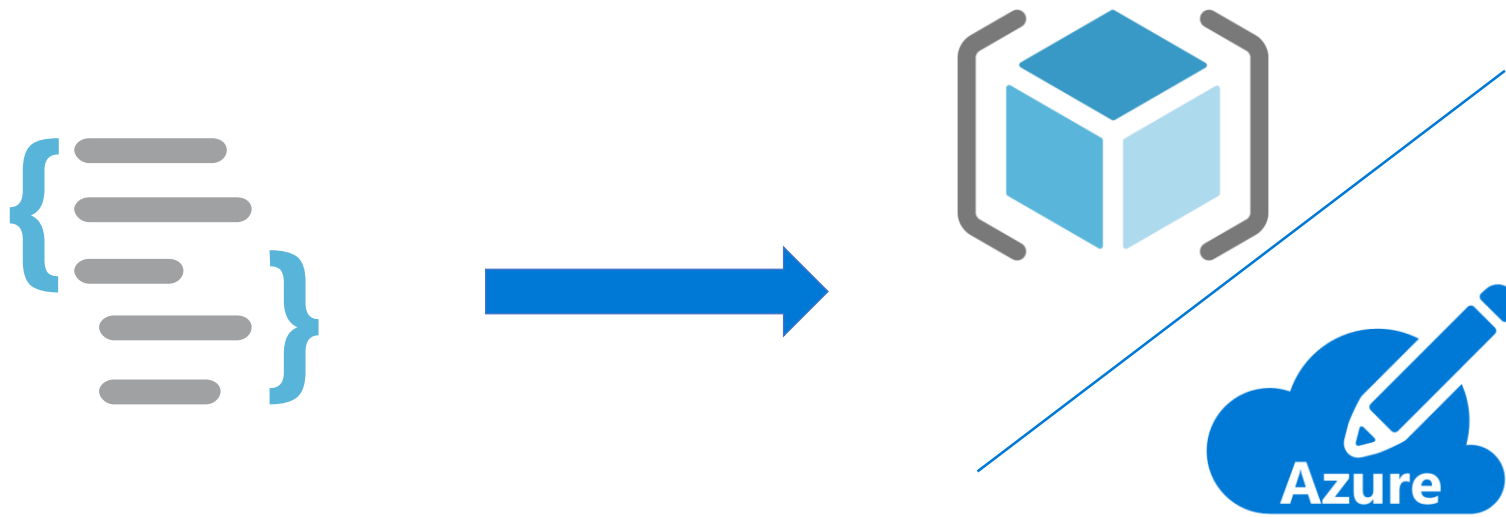
More convenient ways...

```
var azure = Azure
    .Configure()
    .Authenticate(SdkContext.AzureCredentialsFactory.FromFile(/* .. */));
```

```
var resourceGroup = azure.ResourceGroups.Define("myGroup")
    .WithRegion(Region.WestEurope)
    .Create();
```

```
Var sqlServer = azure.SqlServers.Define("UniqueServerName")
    .WithRegion(Region.WestEurope)
    .WithNewResourceGroup("myGroup")
    .WithAdministratorLogin("UserName")
    .WithAdministratorPassword("Password")
    .WithNewFirewallRule(startAddress, endAddress)
    .Create();
```

ARM Templates



Template Structure

```
{  
  "$schema": "...",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
  },  
  "variables": {  
  },  
  "resources": [  
  ],  
  "outputs": {  
  }  
}
```

Parameters

```
"parameters": {  
  "anyParameter": {  
    "type": "string | int | array | object | securestring | secureobject | ...",  
    "minLength": int,  
    "maxLength": int  
  }  
}
```

Variables

```
"variables": {  
  "key": "value"  
}
```

Parameters

```
"parameters": {  
  "environment": {  
    "type": "string",  
    "allowedValues": [  
      "test", "prod"  
    ]  
  }  
}
```

Variables

```
"variables": {  
    "storageAccountName": "[concat('mystorage', parameters('environment'))]"  
}
```


Resources

```
{  
  "type": "Microsoft.Storage/storageAccounts",  
  "name": "[variables('storageAccountName')]",  
  "apiVersion": "2016-01-01",  
  "location": "[resourceGroup().location]",  
  "kind": "Storage",  
  "sku": {  
    "name": "Standard_LRS"  
  }  
}
```

Subresources

```
{  
  "type": "Microsoft.ServiceBus/Namespaces/AuthorizationRules",  
  "name": "namespaceName/ruleName",  
  "apiVersion": "2017-04-01",  
  "location": "[resourceGroup().location]",  
  "properties": {  
    "rights": [  
      "Send"  
    ]  
  },  
  "dependsOn": [  
    "[resourceId('Microsoft.ServiceBus/namespaces', 'namespaceName')]"  
  ]  
}
```

Outputs

```
"outputs": {  
  "storageAccountName ": {  
    "type": "string",  
    "value": "[variables(storageAccountName ')]"  
  }  
}
```

Functions

parameters()

variables()

resourceGroup()

subscription()

resourceId()

concat()

listKeys()

and many, many more...



Good thing there
is a **reference** for
that as well!

<https://docs.microsoft.com/en-us/azure/azure-resource-manager/resource-group-template-functions>

Parameter files

```
{  
  "$schema": "...",  
  "contentVersion": "1.0.0.0",  
  "parameters": {  
    " myParameter ": {  
      "value": "someValue"  
    }  
  }  
}
```

Parameter files

```
{
  "$schema": "...",
  "contentVersion": "1.0.0.0",
  "parameters": {
    "EmailSendgridApiKey": {
      "reference": {
        "keyVault": {
          "id": "/subscriptions/{guid}/resourceGroups/{rgName}/providers/Microsoft.KeyVault/vaults/{vaultName}"
        },
        "secretName": "EmailSendgridApiKey"
      }
    }
  }
}
```

More advanced options

- Subscription level templates
- Linked templates
- Reverse engineering a template
- Resource explorer



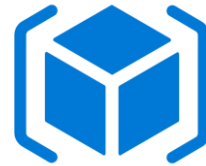
MORE DEMOS,
PLEASE!

SPEED:
120



Test

Pipeline has write access
Team has write access



Acceptance

Pipeline has write access
Team has read access



Production

Pipeline has write access
Team can get access



Finally!

Let's Talk Policy

Azure Policy

With Azure Policy you can ...

- Audit
- Deny
- Augment

any change going through ARM

Azure Policy - definition

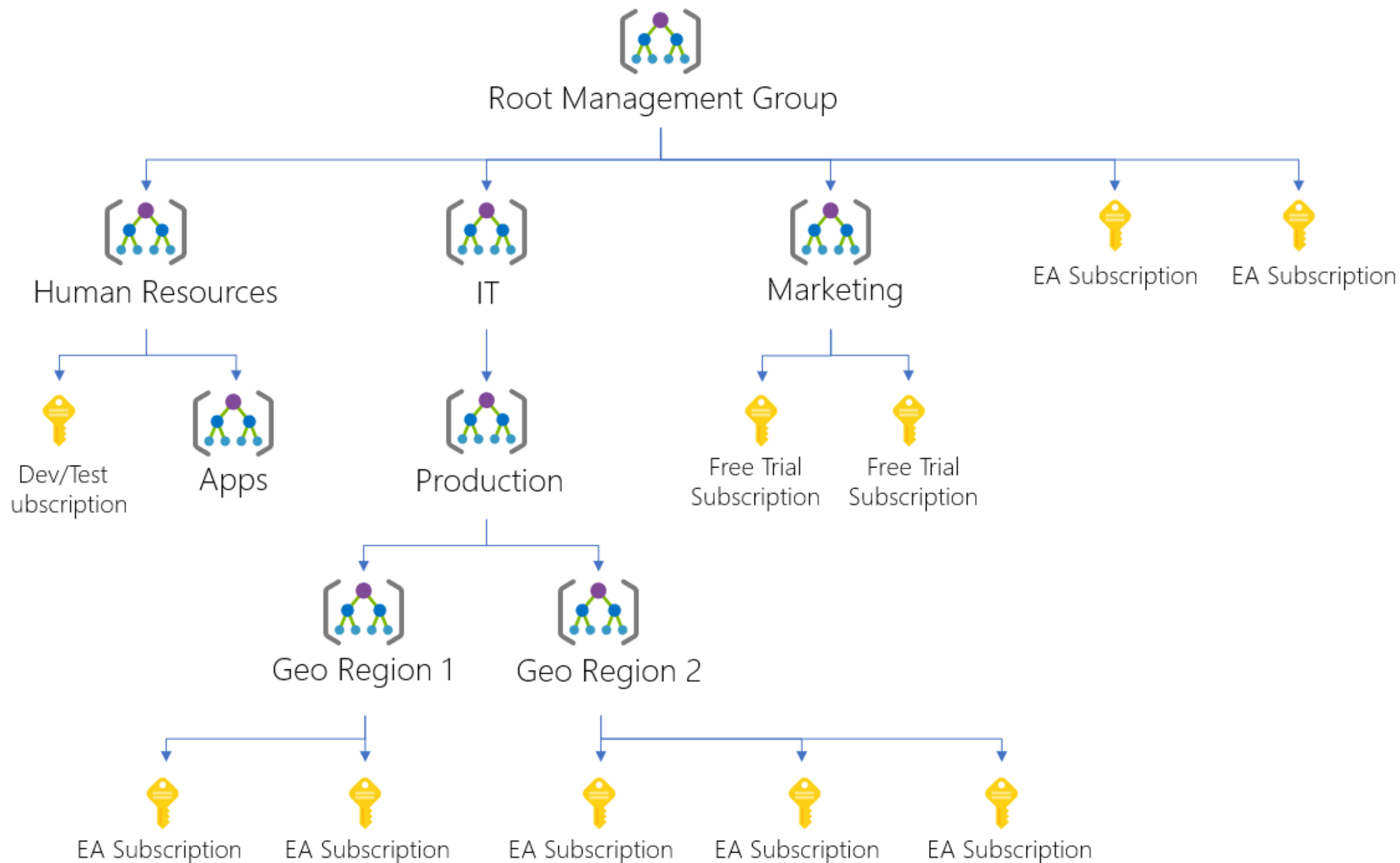
```
{  
  "properties": {  
    "displayName": "... VMs ... maximum password age of 70 days",  
    "policyType": "BuiltIn",  
    "mode": "All",  
    "description": "...",  
    "policyRule": { ... }  
  },  
  "type": "Microsoft.Authorization/policyDefinitions",  
  "name": "24dde96d-f0b1-425e-884f-4a1421e2dcdc"  
}
```

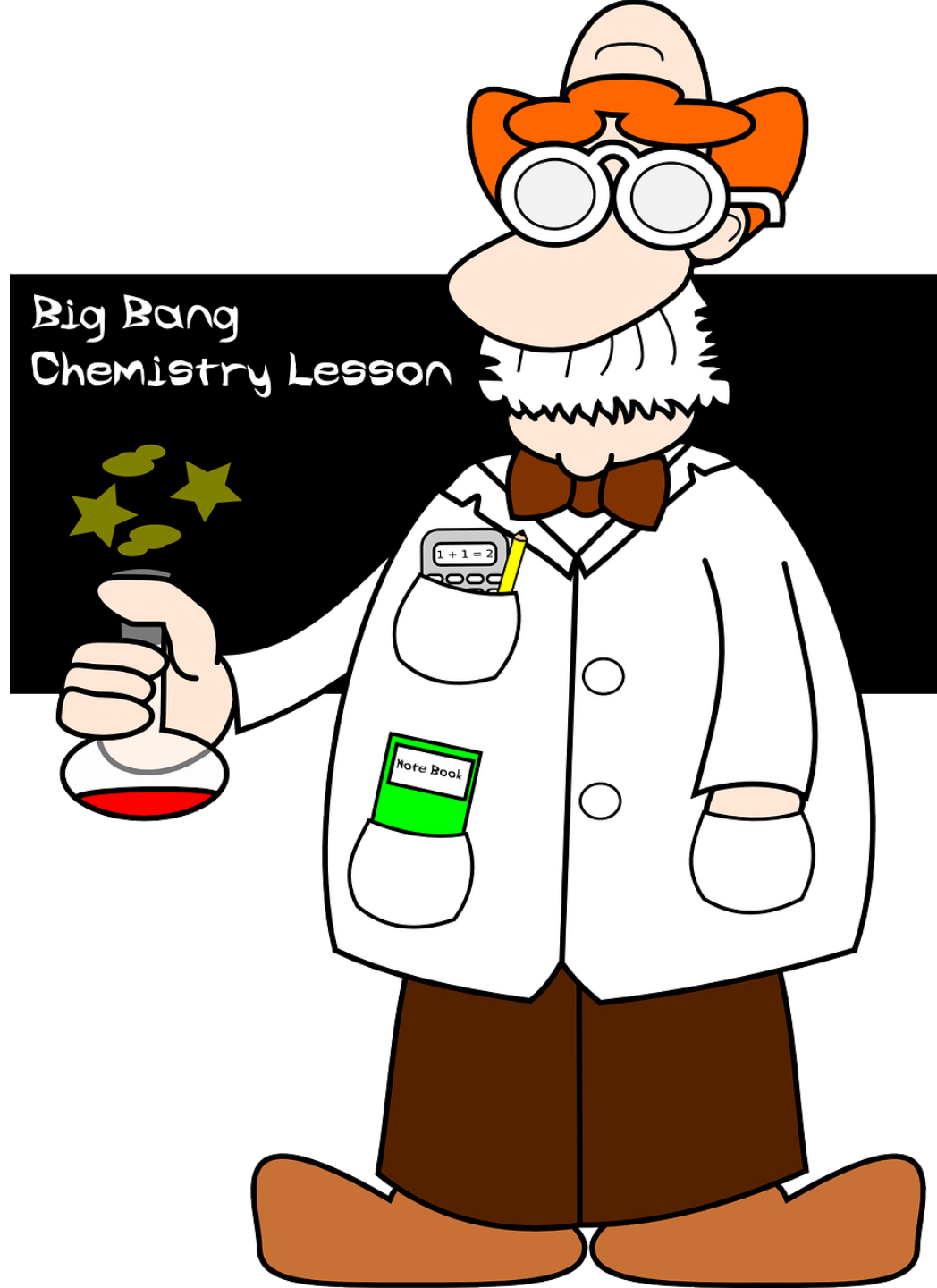
Azure Policy - definition

```
{  
  "properties": {  
    "displayName": "... VMs ... maximum password age of 70 days",  
    "policyType": "BuiltIn",  
    "mode": "All",  
    "description": "...",  
    "policyRule": { ... }  
  },  
  "type": "Microsoft.Authorization/policyDefinitions",  
  "name": "24dde96d-f0b1-425e-884f-4a1421e2dcdc"  
}
```

Azure Policy - definition

```
"policyRule": {  
  "if": {  
    "field": "type",  
    "equals": "Microsoft.classicStorage/storageAccounts"  
  },  
  "then": {  
    "effect": "audit"  
  }  
}
```





LET'S DO A
FINAL DEMO
WITH SOME
JSON FILES



DO TRY THIS **AT HOME!**

HENRY BEEN

Independent Devops & Azure Architect

E: henry@azurespecialist.nl

T: [@henry_been](https://twitter.com/henry_been)

L: [linkedin.com/in/henrybeen](https://www.linkedin.com/in/henrybeen)

W: henrybeen.nl



Questions?

Now is the time!

A photograph of a white CRT computer monitor lying on its side on a grey surface. A wooden baseball bat is stuck through the center of the monitor's screen, which is shattered and cracked. The background is a plain, light-colored wall.

DO TRY THIS **AT HOME!**

HENRY BEEN

Independent Devops & Azure Architect

E: henry@azurespecialist.nl

T: [@henry_been](https://twitter.com/henry_been)

L: [linkedin.com/in/henrybeen](https://www.linkedin.com/in/henrybeen)

W: henrybeen.nl