

Ein Wegweiser durch den Automation Dschungel



Eric Berg
Azure Lead Architekt
COMPAREX AG

MVP Cloud and Datacenter Management

Eric Berg



Lead IT-Architekt – Team Azure / Team Modern Workplace



Azure, Datacenter and Modern Workplace



Azure, System Center, Windows Server and Client



info@ericberg.de



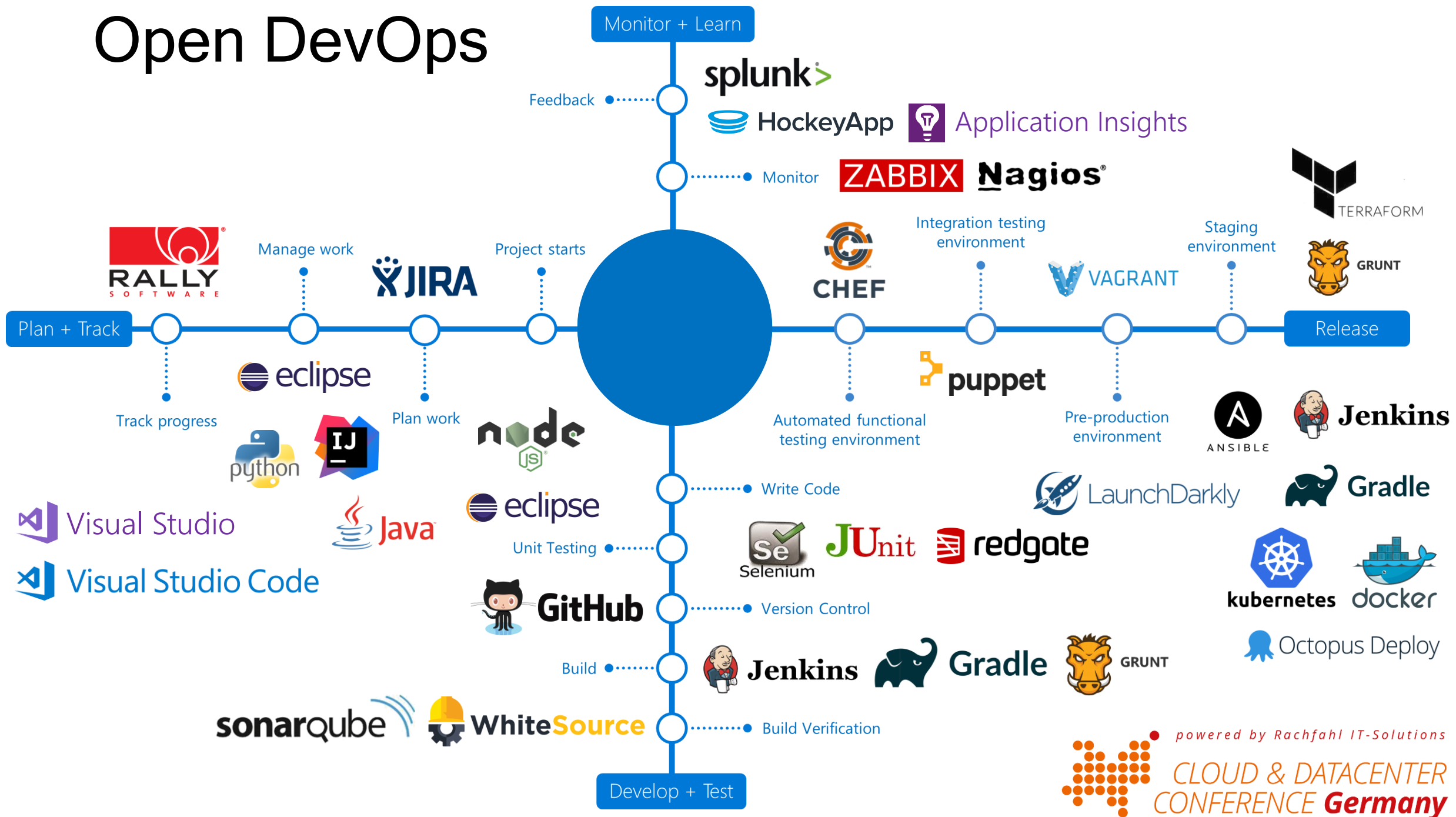
[@ericberg_de](https://twitter.com/ericberg_de) | [@GeekZeugs](https://twitter.com/GeekZeugs)



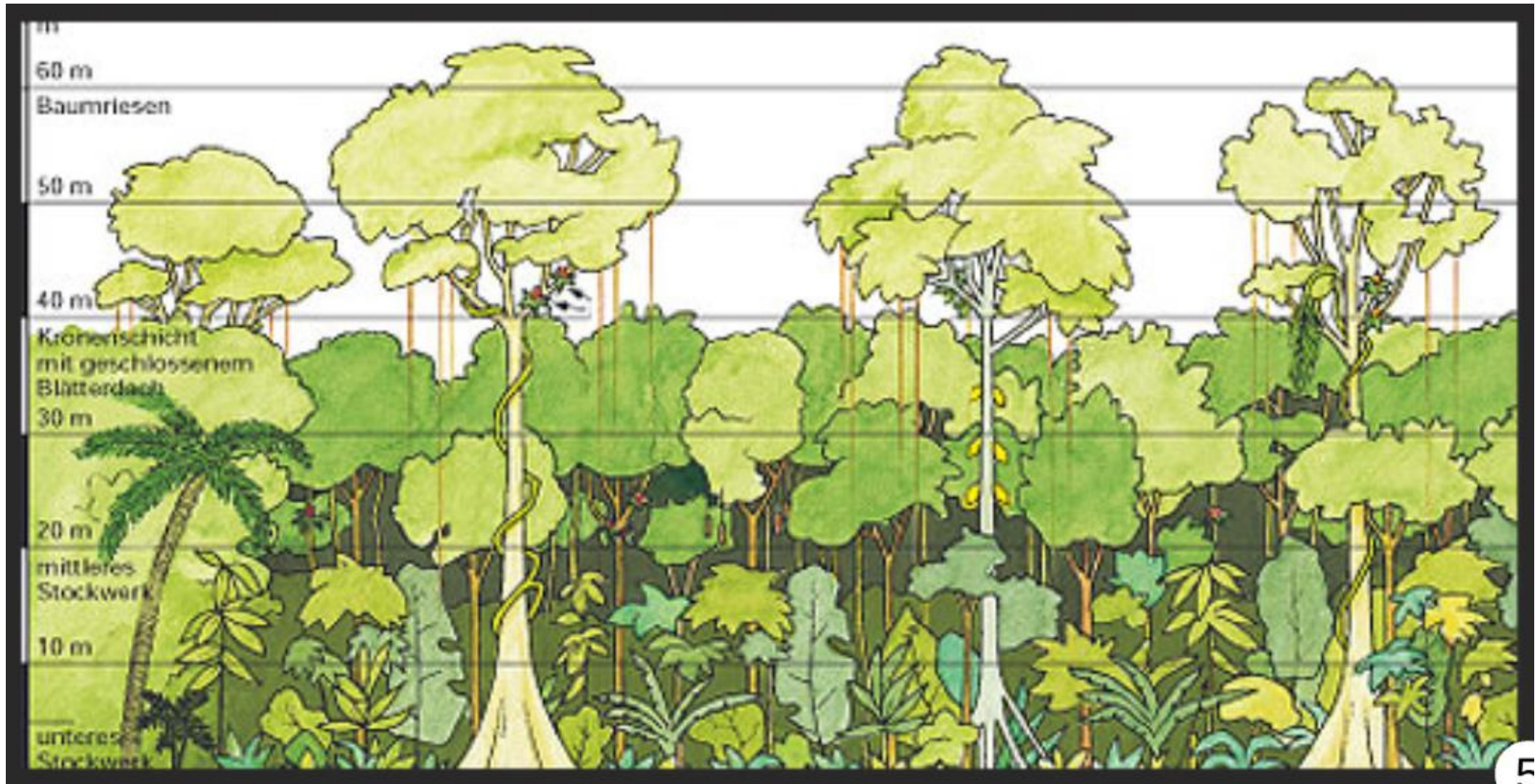
www.ericberg.de | www.geekzeugs.de



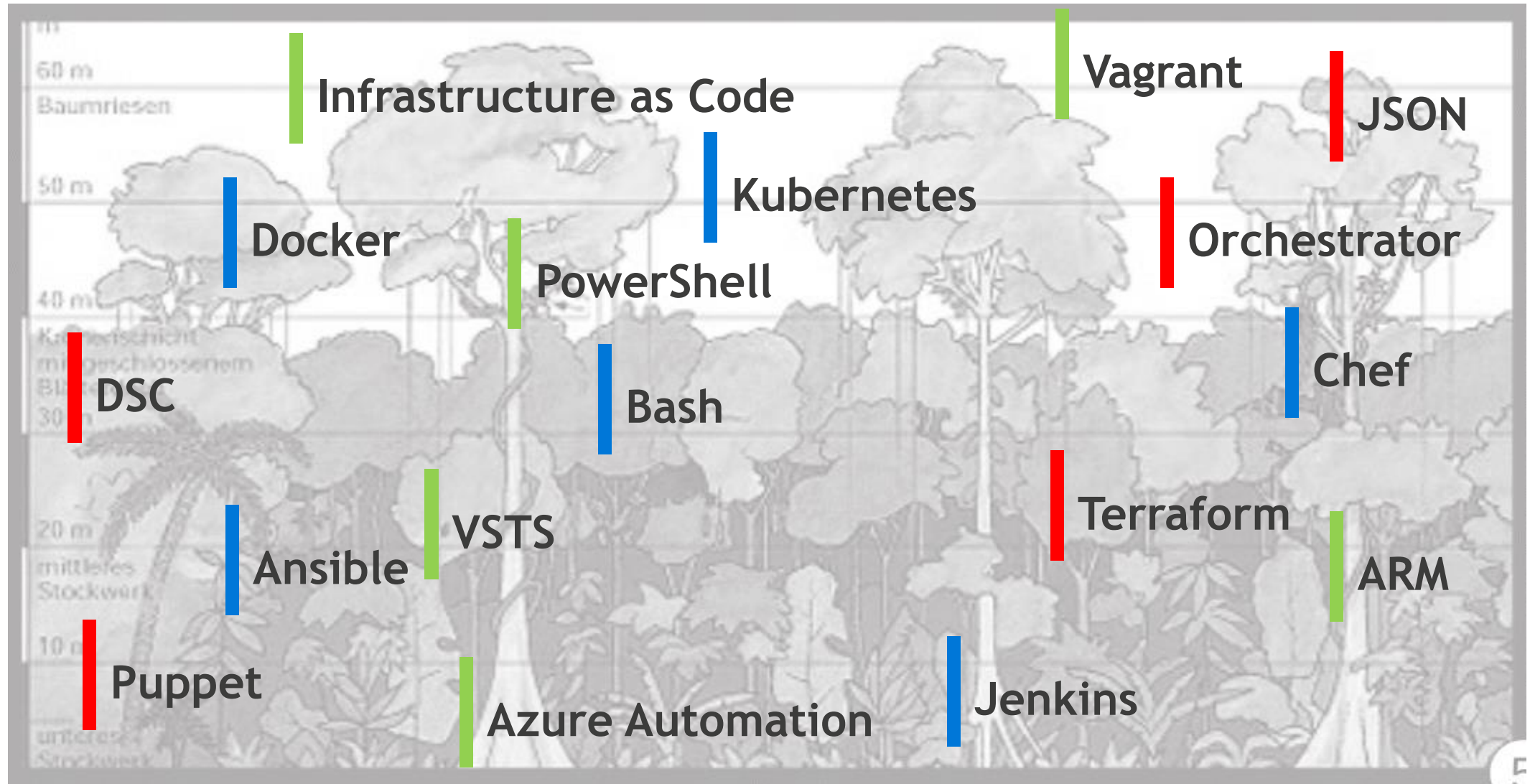
Open DevOps



Dschungel

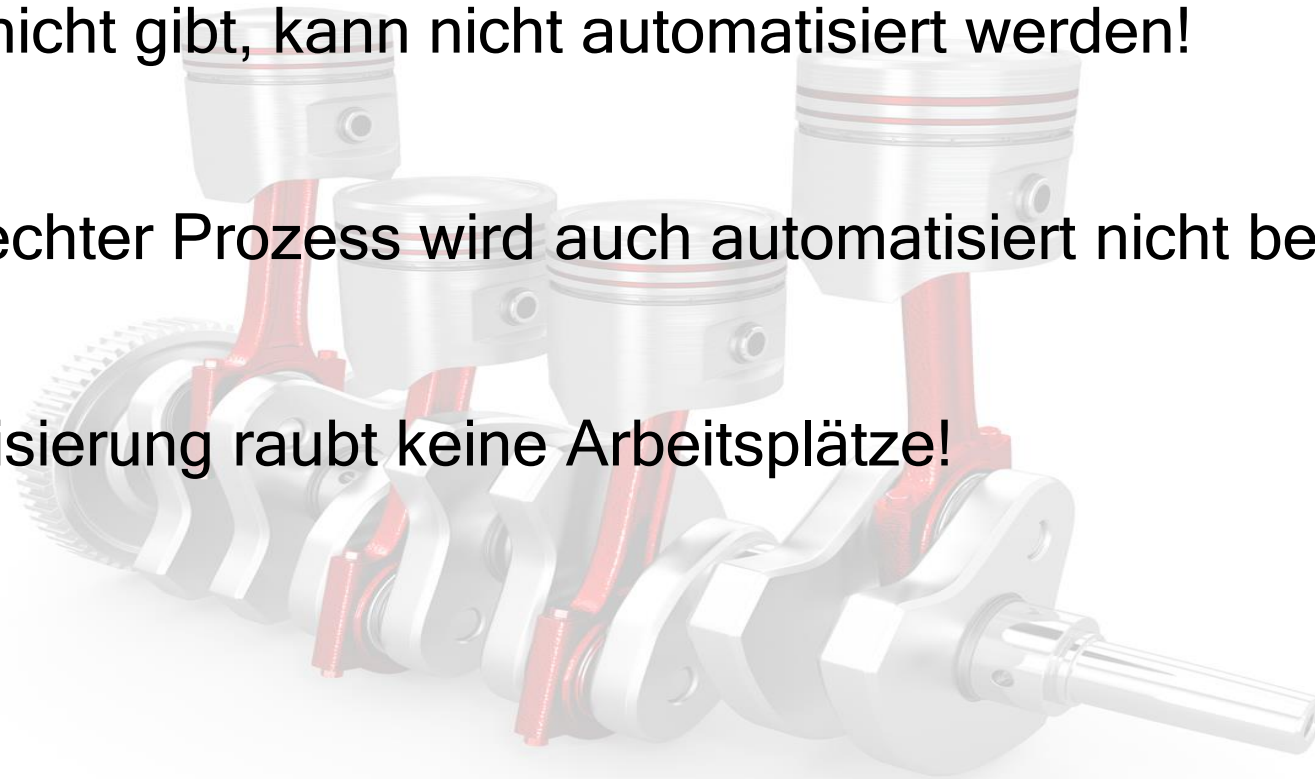


Automation Dschungel



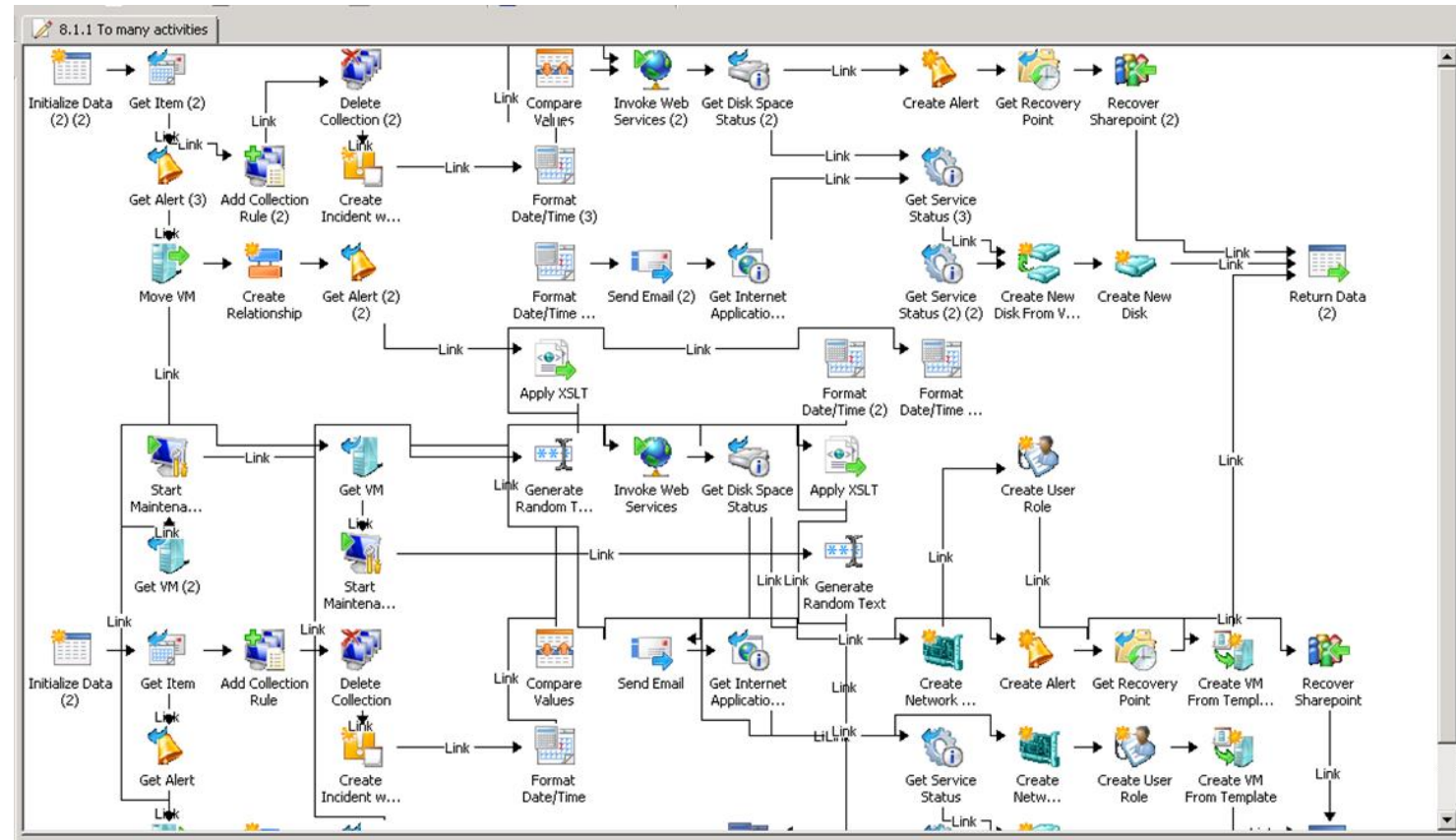
Automation - Fakten

- FAKT I
 - Automatisierung erfolgt oftmals aus den falschen Gründen!
- FAKT II
 - Was es nicht gibt, kann nicht automatisiert werden!
- FAKT III
 - Ein schlechter Prozess wird auch automatisiert nicht besser!
- FAKT IV
 - Automatisierung raubt keine Arbeitsplätze!



Automation - Fallen

- Falle 1
 - Das richtige Werkzeug
- Falle 2
 - Alles auf einmal
- Falle 3
 - Falsche Prios
- Falle 4
 - Fehlende Standards
- Falle 5
 - Mitarbeiter



Automation - Tools und Frameworks

Automation

Workflow Automation

- PowerShell
- Bash
- Au2mator
- Azure Automation

Configuration Management

- Puppet
- Chef
- Ansible
- DSC

Containerization Virtualization

- Docker
- Kubernetes
- Vagrant

PowerShell

- Sequentielle Abarbeitung
- Windows und Linux Support
- Vielzahl cmdlets
- Error-Handling erforderlich
- Bereitstellung + Konfiguration

```
#2. Check location
if(Check-AzureRmLocation -LocationName $LocationName)
{
    #3. Check resource group, if not, created it
    if(Check-AzureRmResourceGroup -LocationName $LocationName)
    {
        #4. Check VM images
        Write-Host "Check VM images $SkusName" -ForegroundColor Green
        If(Get-AzureRMVMImageSku -Location $LocationName -SkuName $SkusName)
        {
            #5. Check VM
            If(Get-AzureRmVM -Name $VMName -ResourceGroup $ResourceGroupName)
            {
                Write-Host -ForegroundColor Red "VM $VMName already exists"
            }
            else{
                #6. Check VM Size
                Write-Host "check VM Size $VMSizeName" -ForegroundColor Green
                If(Get-AzureRmVMSize -Location $LocationName -SizeName $VMSizeName)
                {
                    #7. Create a storage account
                    $BlobURL = AutoGenerate-AzureRmStorageAccountName
                    If($BlobURL){
                        #8. Create a network interface
                    }
                }
            }
        }
    }
}
```

PowerShell DSC

- Deskriptive Zieldefinition
- DSC Provider definieren
Möglichkeiten
- Konfigurations-Management

```
# The Node statement specifies which node this configuration applies to
Node 'AzureRMVM' {

    # The first resource block ensures the WebServer feature is installed
    WindowsFeature WebServer {
        Ensure = "Present"
        Name   = "Web-Server"
    }

    # The second resource block ensures the website content is present
    File WebsiteContent {
        Ensure = 'Present'
        SourcePath = 'c:\test\index.html'
        DestinationPath = 'c:\inetpub\wwwroot'
    }
}
```

ARM / JSON

- Azure Resource Manager
- Deskriptive Zieldefinition
- Parallele Verarbeitung
- Erweiterbar durch Extensions
- Nativ in Azure vorhanden

```
"apiVersion": "2016-04-30-preview",
"type": "Microsoft.Compute/virtualMachines",
"name": "myVM",
"location": "[resourceGroup().location]",
"dependsOn": [
  "[resourceId('Microsoft.Network/networkInterfaces/', 'myNic')]"
],
"properties": {
  "hardwareProfile": { "vmSize": "Standard_DS1" },
  "osProfile": {
    "computerName": "myVM",
    "adminUsername": "[parameters('adminUsername')]",
    "adminPassword": "[parameters('adminPassword')]"
  },
  "storageProfile": {
    "imageReference": {
```


Terraform

- Infrastructure as Code
- Multi-Cloud
- Integriert in Azure Cloud Shell
- Kein Konfig-Management
- Integration mit anderen Tools möglich

```
# create virtual machine
resource "azurerm_virtual_machine" "CDCTFVM" {
  name = "CDCTFVM"
  location = "West Europe"
  resource_group_name = "${azurerm_resource_group.name}"
  network_interface_ids = ["${azurerm_network_interface.id}"]
  vm_size = "Standard_A2"

  storage_image_reference {
    publisher = "Canonical"
    offer = "UbuntuServer"
    sku = "14.04.2-LTS"
    version = "latest"
  }
}
```

Puppet

- Agent basiert
- Pull Agents
- Konfigurations-Management
- Facts vs. Katalog
- Desired State (Ruby)

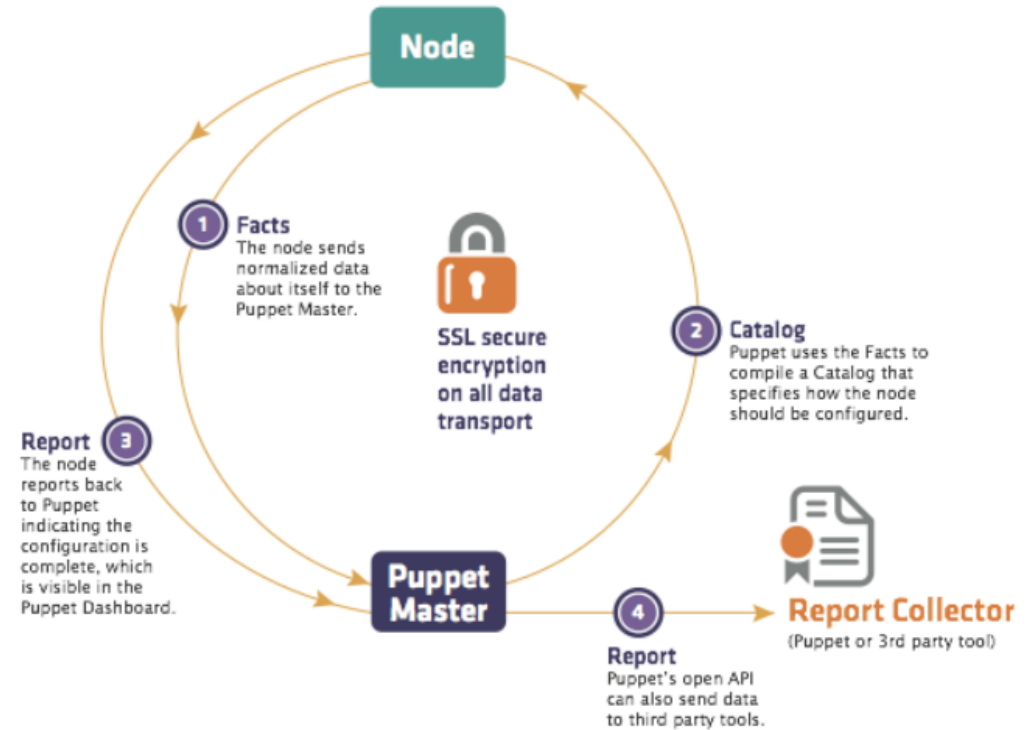
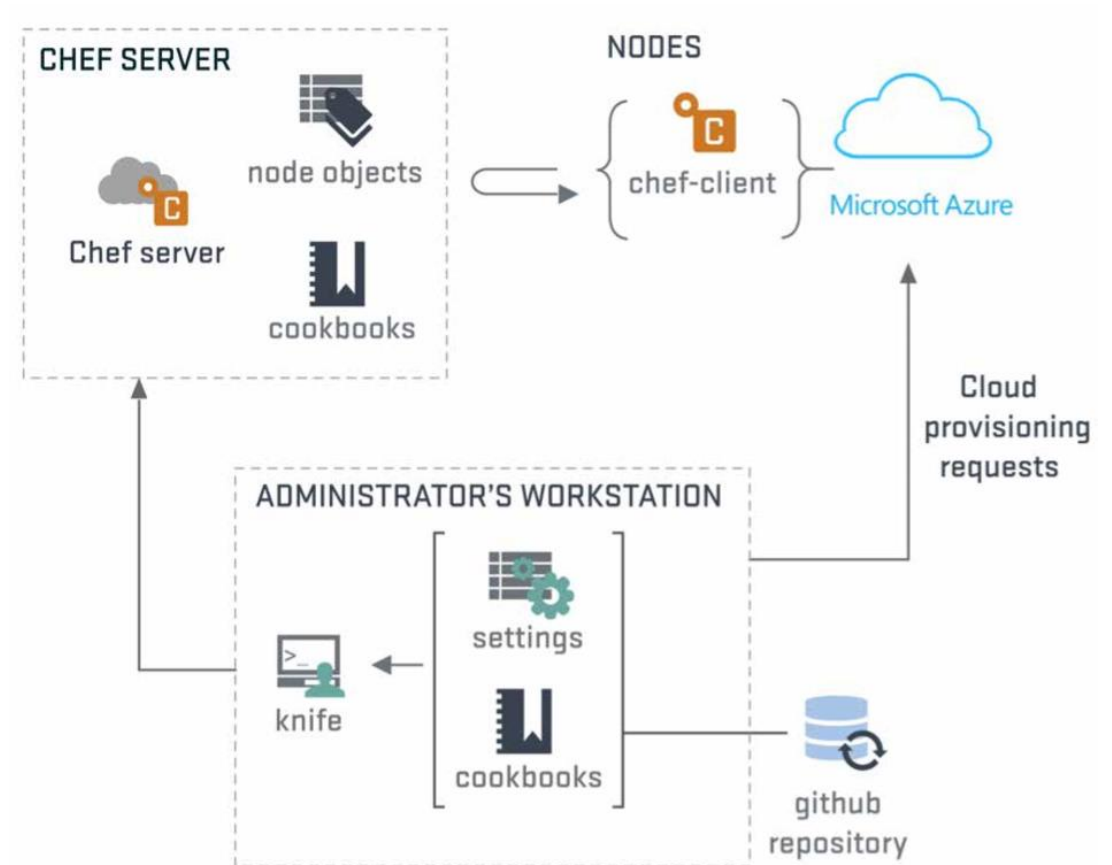


FIGURE 4.1.5 Puppet Architecture

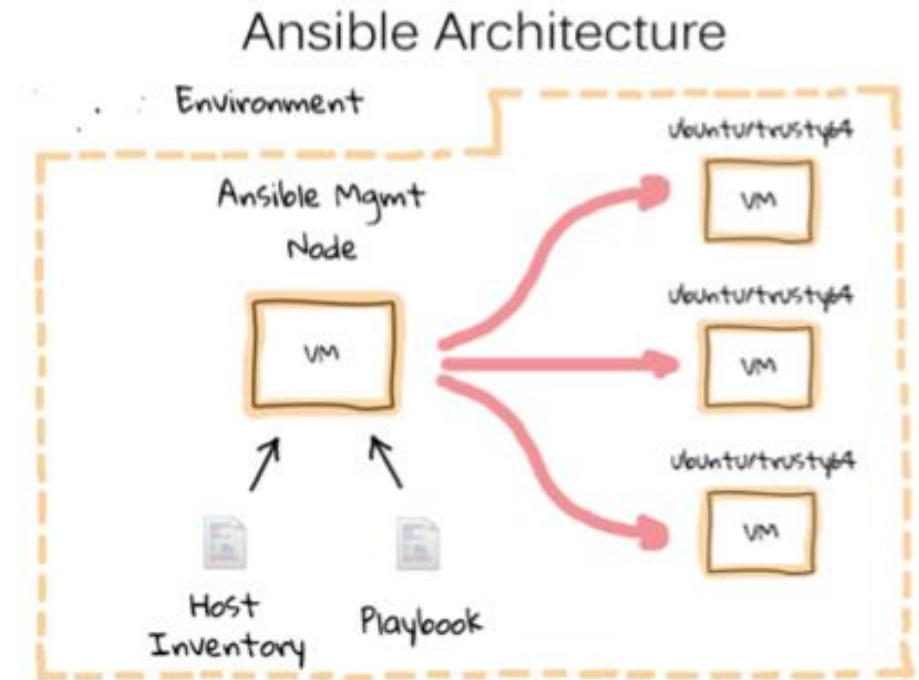
Chef

- Agent basiert
- Pull Agents
- Konfigurations-Management
- Cookbooks (Ruby)
- Abläufe

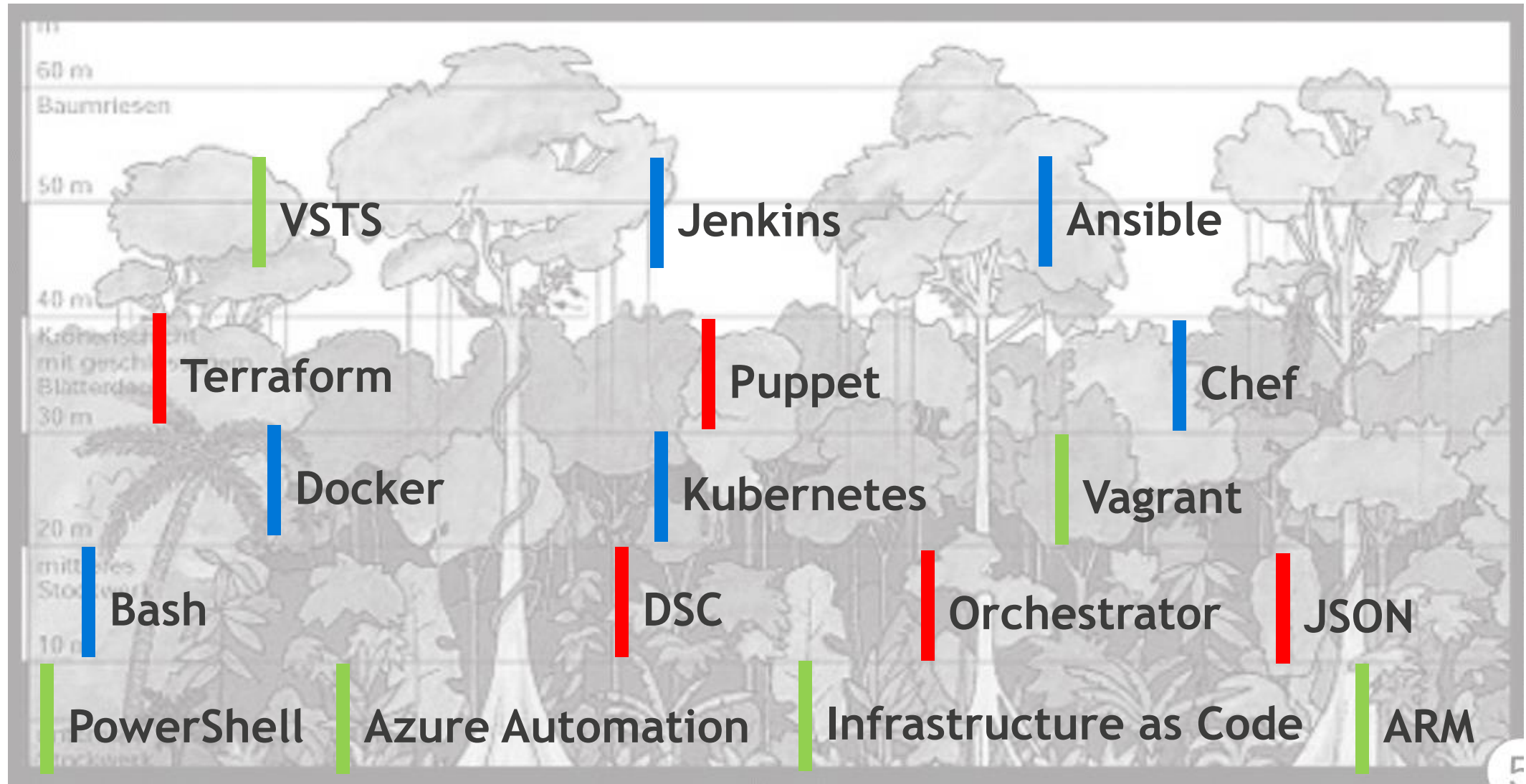


Ansible

- Bereits in Azure Cloud Shell installiert
- Agent-less
- Push basiert
- SSH oder WinRM
- small / medium
- Python



Automation Dschungel



Danke an unsere Partner!

Platinum Sponsor



Gold Sponsoren



Danke an unsere Partner!

Gold Sponsoren

