



# Empowering Efficiency

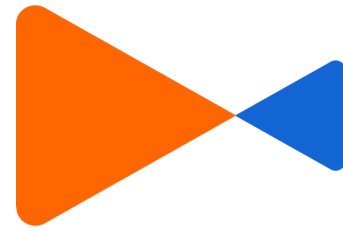
The benefits of Automation and how to achieve it with Azure

Dimitris Pantazis

Dear Global Azure Athens  
2023 sponsors,  
your support made all the  
difference — **thank you!**



#GlobalAzureAthens



kaizen  
GAMING



Microsoft

Info Quest  
TECHNOLOGIES



BlueStream  
SOLUTIONS



Code.Hub

SIGNAL™

Cloud & Automation  
Architect  
**UniSystems**

Microsoft Certified  
Trainer



Dimitris Pantazis





# Agenda

Steps in your  
Automation journey

Azure Services and  
Automation tools

#GlobalAzureAthens

# Steps in your Automation Journey



#GlobalAzure

# My own experience...

Manual  
Processes  
in Word  
docs

Time  
consuming

Prone to  
errors

Not so  
smart

# BORING!!!

# ...and how I worked

- Started to challenge my colleagues
- Picked trivial activities that could be automated easily
- PowerShell ♥♥♥
- Then cloud came into my life

# Why should I consider Automation?

- Automation = Code
- Consistency
- Repeatability
- New skills
- FUN!!



# Objections to Automation (and how to overcome)

- Don't know how to start
- Need to deliver my project on time
- Learning curve
- If all tasks are automated what happens with my job?
- The rest of the team is not really into this

# How to start...?

- Look for repetitive manual tasks in your routine
- Search on the Internet (e.g. Github)
- Read some code and try to get familiar with the technology
- Try! Try! Try!



# How to gain others' attention?

- Look for people that share similar interests
- Tell them about the exciting tools you build
- Ask them to start using your tools and listen to their feedback
- Earn management buy-in
- Expand in more projects

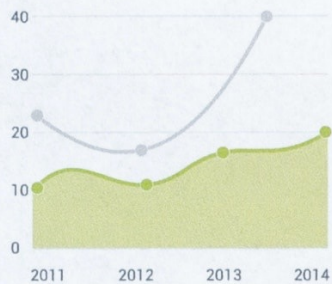


# Establish clear metrics that show your work

## Examples:

- Saved 1hr of daily manual work
- I can apply changes to 20 servers/hr instead of 5
- Errors reduced by 50%

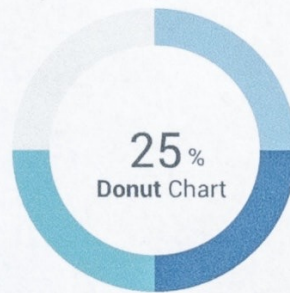
Area Chart



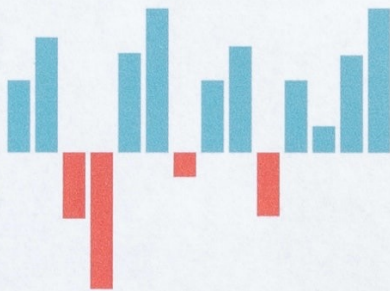
Bar Chart



Donut Chart



Bar Chart



Pie Chart





# Make sure to...

- Orientate
- Test and measure
- Stay up to date
- Secure

Automate with  
Purpose!



#GlobalAzureAthens

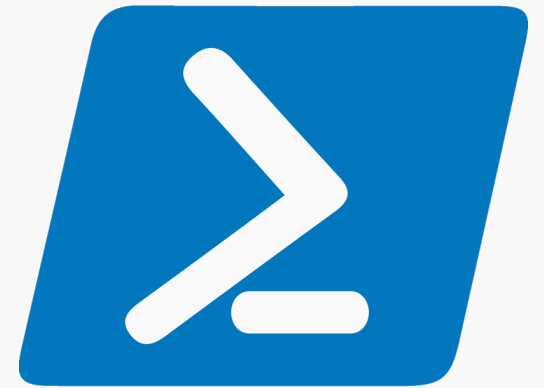
# Azure Services and Automation tools



#GlobalAzure

# Az PowerShell / Az CLI

- Imperative syntax
- Used in scripts that execute commands in specific order



# Az PowerShell

```
PS /home/dimitris> get-azresourcegroup | select resourcegroupname,location
```

ResourceGroupName	Location
dpantazblog	westeurope
cloud-shell-rg	westeurope
myorg-storage	westeurope

# Az CLI

```
PS /home/dimitris> az group list --output table
```

Name	Location	Status
dpantazblog	westeurope	Succeeded
cloud-shell-rg	westeurope	Succeeded
myorg-storage	westeurope	Succeeded

# Infra as Code

- Declarative syntax
- Store in Git
- Repo becomes infra source of truth
- Pipelines to apply changes
- Bicep / Terraform





# Bicep

```
120  ✓ resource pip 'Microsoft.Network/publicIPAddresses@2021-02-01' = {  
121      name: publicIpName  
122      location: location  
123  ✓   sku: {  
124      //    name: 'Standard'  
125      }  
126  ✓   properties: {}  
127  //    publicIPAllocationMethod: 'Static'  
128      }  
129  }
```

# Terraform

```
43  # Create public IPs  
44  ✓ resource "azurerm_public_ip" "pip" {  
45      name                = var.pipname  
46      location            = var.location  
47      resource_group_name = azurerm_resource_group.rg.name  
48      allocation_method   = "Static"  
49      sku                 = "Standard"  
50  }  
51
```

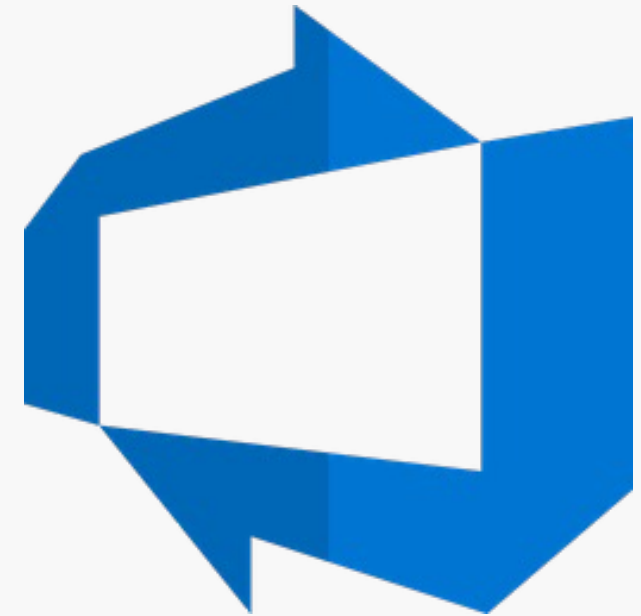
# Azure Automation

- Cloud-based automation
- PowerShell / Python
- Configuration Management
- Automated patching



# Azure DevOps

- Complete CI/CD solution
- Code / Build / Test / Deploy
- Project Management
- Store your code in repos
- Create pipelines to deploy automation

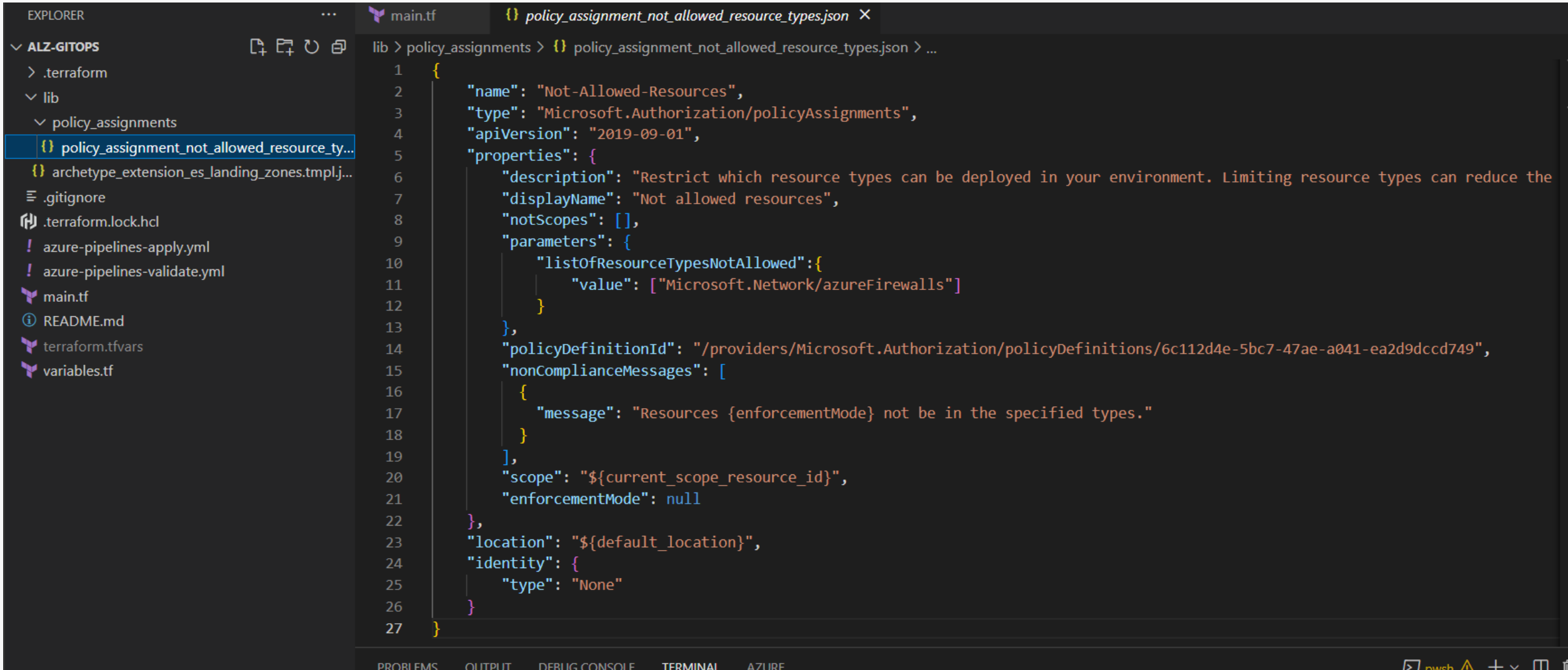


# Azure Policy

- Compliance
- Consistency
- Deploy If Not Exists / Modify
- Policy as Code
- Use pipelines to test and then deploy
- Azure Landing Zone Policies



# Policy as Code



The screenshot shows the Visual Studio Code interface with a Terraform project. The Explorer sidebar on the left shows the file structure: `ALZ-GITOPS` (expanded) contains `.terraform`, `lib` (expanded), and `policy_assignments`. The `policy_assignments` folder contains the file `policy_assignment_not_allowed_resource_types.json`, which is currently selected and open in the main editor. The breadcrumb navigation at the top of the editor shows the path: `lib > policy_assignments > policy_assignment_not_allowed_resource_types.json > ...`. The main editor displays the JSON configuration for the policy assignment, with line numbers 1 through 27 on the left margin. The JSON content is as follows:

```
1  {
2    "name": "Not-Allowed-Resources",
3    "type": "Microsoft.Authorization/policyAssignments",
4    "apiVersion": "2019-09-01",
5    "properties": {
6      "description": "Restrict which resource types can be deployed in your environment. Limiting resource types can reduce the",
7      "displayName": "Not allowed resources",
8      "notScopes": [],
9      "parameters": {
10       "listOfResourceTypesNotAllowed": {
11         "value": ["Microsoft.Network/azureFirewalls"]
12       }
13     },
14    "policyDefinitionId": "/providers/Microsoft.Authorization/policyDefinitions/6c112d4e-5bc7-47ae-a041-ea2d9dccc749",
15    "nonComplianceMessages": [
16      {
17        "message": "Resources {enforcementMode} not be in the specified types."
18      }
19    ],
20    "scope": "${current_scope_resource_id}",
21    "enforcementMode": null
22  },
23  "location": "${default_location}",
24  "identity": {
25    "type": "None"
26  }
27 }
```

At the bottom of the window, there are tabs for `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, `TERMINAL`, and `AZURE`. The `TERMINAL` tab is active, showing a `pwsh` prompt.

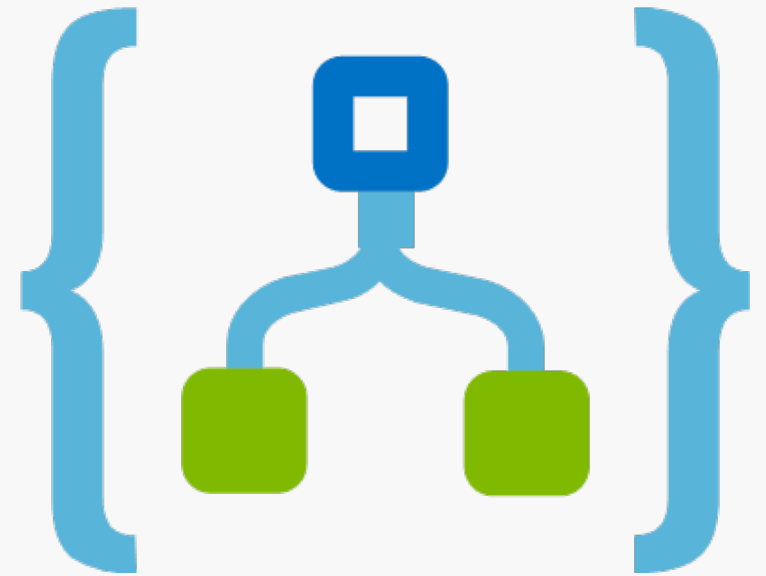
# Azure Functions

- Small pieces of code
- Run scripts in the cloud
- Serverless
- Multiple programming languages
- Input / Process / Output



# Azure Logic Apps

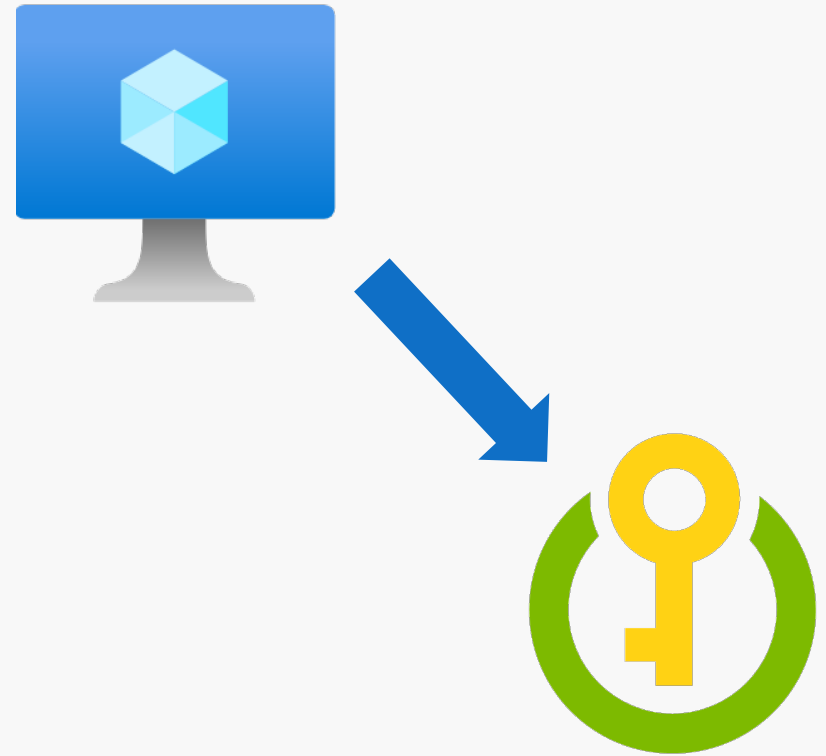
- Workflow based automation
- Apply business logic
- Endlessly integrate with internal/external systems
- Trigger based schedules/events or manually

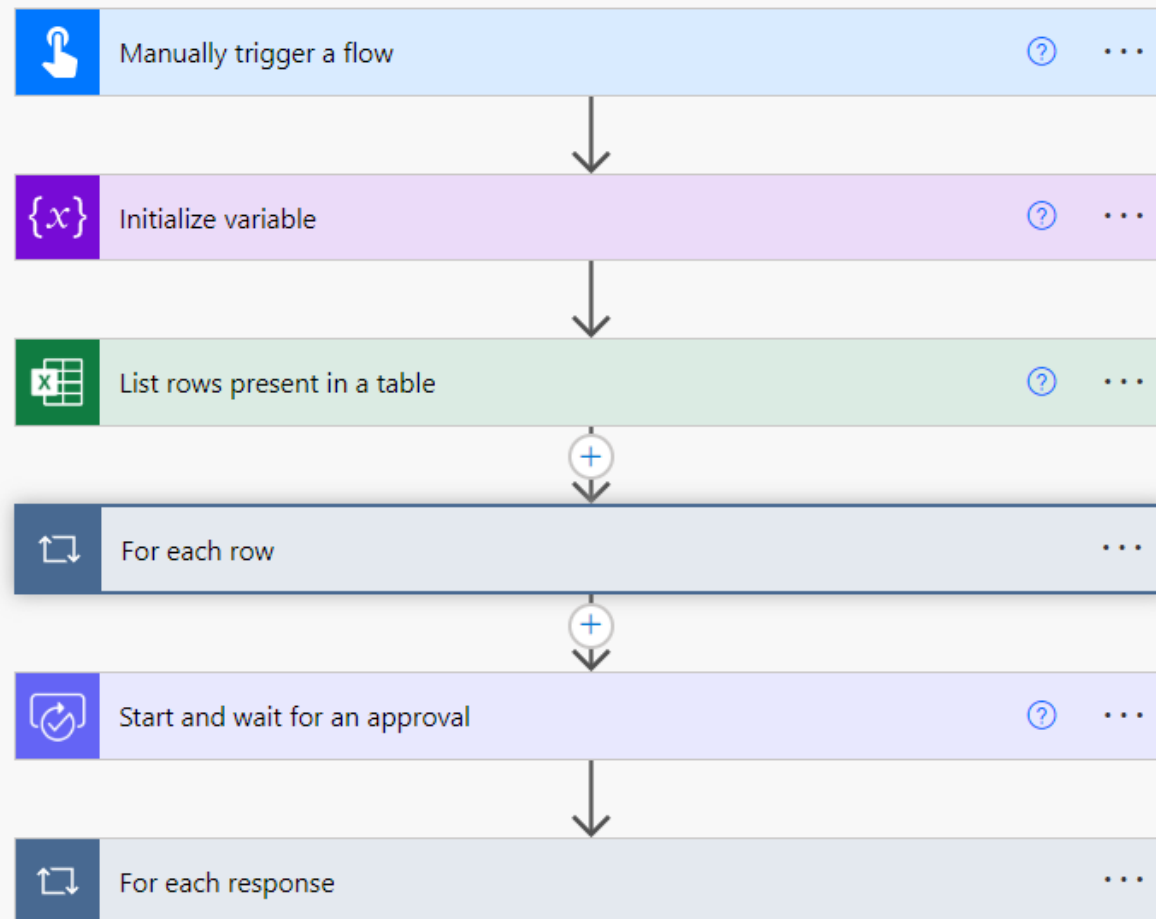




# Managed Identities

- System-assigned
- User-assigned
- Allow Azure Services to authenticate against other services without passwords
- RBAC compatible





+ New step

Save

# Azure Monitor

- Configure alerts
- Create tickets in your ITSM
- Respond to alerts with automation
- Autoscale
- Call a function or an Azure Automation runbook



# Azure Site Recovery

- Automate your disaster recovery
- Recovery plans
- Failover your VMs in the right order
- Save costs and make your DR process consistent



# Microsoft Sentinel

- SOAR capabilities
- Automate your response to security incidents
- Create automation rules
- Execute playbooks
- Huge collection available on Github



#GlobalAzureAthens

# Closing



#GlobalAzure

# Final thoughts

- Find the ideal starting point
- Identify the tooling that fits better
- Store code in repo
- Validate
- Continuously Improve



# Be the Automation Hero...

... and help create more heroes  
in your team







<http://flymetothecloud.com>



/dpantaz



/dpantaz



@CloudTechnologyTuesdays



Thank you!



Please evaluate !



A big **thank you** to our sponsors!



Microsoft

InfoQuest  
TECHNOLOGIES



CUBE



BlueStream  
SOLUTIONS



Code.Hub

SIGNAL

<https://bit.ly/GA23Evaluation>

#GlobalAzureAthens