

Mastering Azure Resources

Marcel Zehner | itnetX
Corporate Ambassador
Microsoft Regional Director (RD)
Microsoft Most Valuable Professional (MVP)



45 minutes only ...

What I will cover

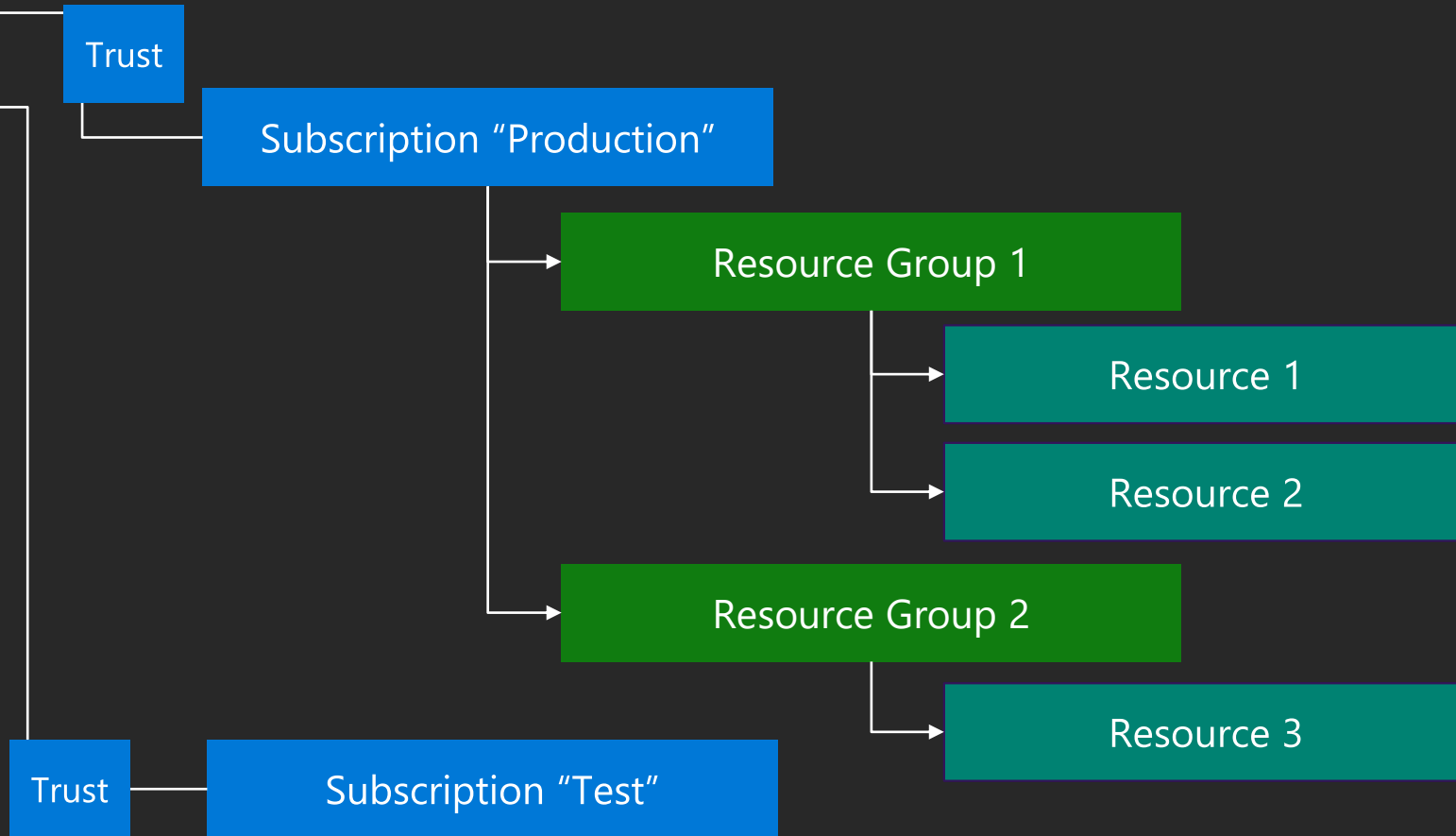
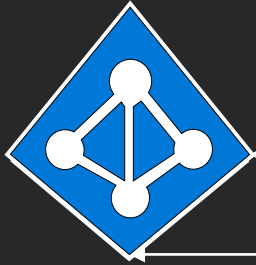
- Focus mainly on the technical part of things
- Foundation to manage Azure resources
 - Azure governance
- Resources provisioning options
- Azure Resource Manager (ARM)

What I will NOT cover

- Azure Service Management (ASM)
- Organizational management
 - EA portal
- Costs management

Hierarchy

Azure AD
marcelzehner.
onmicrosoft.com
marcelzehner.ch



Subscriptions

- First container that is created
 - Global object, not bound to a region
 - Holds resource groups and resources
- Subscription trusts 1 Azure AD tenant
 - Used to assign management permissions to users & groups
- **How many subscriptions?** It depends ...
 - Subscription limits that could be reach?
 - Trusted subscription owner?
 - Need to block specific resource providers?
 - Can RBAC be used to delegate access?

Subscription best practices

- Use as few subscriptions as possible
- Check subscription limits before defining your subscription strategy

Limitations

- Check out this article for limits details
 - <https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits>
- Many default limits can be increased by creating a free support request

Resource Groups

- Every resource belongs to a resource group
 - Container for resources
 - Created in a region
 - Does not mean that all contained resources reside in the same region
 - During deployment, the region can be inherited by resources (not mandatory)
- **How many resource groups?** It depends ...
 - Do resources share the same lifecycle?
 - Are resources managed by the same team?

Resource group best practices

- Create resource groups based on management access first, then lifecycle
- Use appropriate region

Limitations

- Check out this article for limits details
 - <https://docs.microsoft.com/en-us/azure/azure-subscription-service-limits>

Naming

- Naming convention for Azure objects
 - Better transparency
 - Key for automated resource provisioning
- Example naming definition
 - **location-environment-type-name-number**
- Examples
 - Subscription > **tst-sub-playground-01**
 - Resource Group > **weu-prd-rg-webshop-01**
 - Vnet > **weu-prd-vnet-bern-01**

Naming best practices

- Define naming conventions before anything gets deployed

<https://docs.microsoft.com/en-us/azure/architecture/best-practices/naming-conventions>

Limitations

- Length is restricted, varies by entity (resource type), check link
- Some entities need a globally unique name
 - E.g. storage accounts

Tagging

- Logically organize Azure objects
 - Resource groups and resources
- Name/value pairs
- **No inheritance** from resource group to resource
- Example tags
 - Owner: Marcel Zehner
 - Application id: WebShop
 - CostCenter: 145
 - Environment: Production

Tagging best practices

- Use tags for more transparency
- Search for untagged resources on a regular basis and update tags

Limitations

- Tags per RG/resource: 15
- Name length: 512 chars
 - 128 chars for storage accounts
- Value length: 256 chars

RBAC 1/2

- Assign permissions to Azure resources
 - Least privilege model
- Role definitions
 - Set of permissions
 - Can be used in multiple assignments
 - Example: "Reader"
- Role assignments
 - Associate role definitions with an identity at a scope
 - Example: Assign "Reader" to User "Marcel Zehner" on Subscription "Test"
- Permissions are inherited
 - Inheritance **cannot** be blocked

RBAC 2/2

- Three primary role definitions
 - **Owner** > Full access to resource, incl. the right to delegate access
 - **Contributor** > Create and manage resources
 - **Reader** > View-only
- Available for all Azure resources
- Many resource specific role definitions
 - SQL DB Contributor
 - Site Recovery Operator
- Custom role definitions

RBAC best practices

- Least privilege model
- Organize resources to meet access management requirements
- Grant access at resource group level wherever possible
- Prefer existing over custom roles
- Use group instead of user assignments

Limitations

- Inherited permissions cannot be blocked or removed

Locks

- Lock down of subscriptions, resource groups and resources
- Preventing accidental deletion or modification
- 2 lock levels
 - **CanNotDelete** > Read & edit, not delete
 - **ReadOnly** > Read
- Locks are inherited
- The most restrictive lock wins

Locks best practices

- Use locks inheritance where it makes sense
- Configure RBAC first, then add locks where it makes sense

Limitations

- Max. 20 unique locks per scope

Policies 1/2

- Enforce rules and actions
 - To stay compliant (force)
 - Monitor compliance (audit)
- Examples
 - Restrict Azure regions
 - Restrict services
 - Enforce tagging
 - Enforce naming conventions

Policies 2/2

- Policy definitions
 - If/then definitions
 - Some are delivered by Microsoft
 - Own definitions can be created if needed
 - Can use parameters for flexibility
- Initiative definitions
 - Group of multiple policy definitions
- Policy & initiative assignment
 - Assign definitions to a scope
 - Subscription or resource group

Policies best practices

- Always use initiatives, not policies
- Implement at the highest possible level
- Implement in phases: Audit first, then deny
- Try out the new policy console (preview)

Demo

Tagging, RBAC, Locks, Policies

Provisioning 1/2

- Provisioning options
 - Azure Marketplace
 - Azure Service Catalog
 - PowerShell
 - Azure CLI
 - Cloud Shell
 - Azure Resource Manager (ARM) templates
- Manual or automated?
 - Manual > OK for test & play
 - Automated > Preferred for dev, QA and prod

Provisioning 2/2

- ARM templates
 - **Declarative** approach (JSON)
 - Full lifecycle management (create, update, delete)
 - Test inside out in dev, then deploy 1:1 into QA & prod
- Use release pipeline to automate deployments
 - Full technical deployment & testing process incl. approvals
- Integrate into ITSM solutions
 - Company self service catalog
 - Full business processes

Provisioning best practices

- No manual deployments in production
- Use ARM deployments wherever possible
- Use release pipelines

Demo

ARM Templates, Release Pipeline

Recap

- Address Azure governance first, then start deploying
 - Naming conventions
 - Tagging
 - RBAC
 - Locks
 - Policies
 - Etc.
- Deploy new resources into prod by using ARM templates & automation
 - Fewer errors, faster delivery and much better consistency

Please Complete your Session Evaluations

Get your cool IoT Dev Kit!

Fill out your feedback form and turn it in before you leave.



Mastering Azure Resources

Marcel Zehner | itnetX
Corporate Ambassador
Microsoft Regional Director (RD)
Microsoft Most Valuable Professional (MVP)

