



## Azure Infrastructure with Terraform: 1-day Workshop

### Program

#### Day-1

- **Module:** Introduction (1h)
  - What is IAAC (Infrastructure as a Code) and how it differs from other provisioning techniques?
  - What is Terraform and what is the benefits of utilizing it?
- **Module:** Terraform in Detailed (2h)
  - Terraform architecture and components
  - Terraform workflow (init, plan, apply, destroy)
  - Using Terraform in collaborative manner
  - Setting up the demo environment
- **Lab:** Provisioning Azure Resources (5h)
  - Terraform configuration
    - Setup local state file
    - Initialize Azure provider for provisioning resources on the cloud
  - Build Infrastructure
    - Create sample Azure resource group and a keyvault in it
    - Initialize, plan and then apply configuration
    - Inspect the Terraform state and created resources on Azure portal
  - Change Infrastructure
    - Add tags to existing resources and create a secret in the keyvault
    - Inspect the plan output to see how Terraform manages infrastructure changes
  - Destroy Infrastructure



# Deep Network GmbH

Landsbergerstr. 302 | 80687 München | Deutschland +49 (0) 89 6151 5236 | [www.deepnetwork.com](http://www.deepnetwork.com)

- Evaluate the destroy plan and confirm
- Use Variables
  - Setup two different environment with separate backends
  - Create individual variable files for both
  - Create same Azure resources in both environment with different values for the variables
  - Apply both configuration and check the difference on Azure portal
- Use Modules for Common Configurations
  - Create common Azure resources on separate directory
  - Reference this directory as a module for both environment
  - Apply both configuration and check the result with the previous step
- Import existing Azure resources
  - Create resource group on Azure portal
  - Import it into the existing Terraform state via Terraform's import utility
  - Write the corresponding configuration
  - Review the Terraform plan to ensure configuration matches the expected state and infrastructure
  - Apply the configuration to update Terraform state
- Setup Remote State
  - Create storage account on Azure for keeping Terraform state file
  - Initialize the Azure backend with remote state file
  - Create some resources and apply the configuration
  - Setup new workspace within local system or another machine with the same configuration
  - Initialize configuration and check the state