

UNIVERSITÄT MANNHEIM

GREENPEAK PARTNERS

AI DEPLOYMENT AND OPERATIONS

CHAIR OF ARTIFICIAL INTELLIGENCE | PROF. DR. HEINER STUCKENSCHMIDT
GREENPEAK PARTNERS | LINDA HONG

PROJECT SEMINAR

STRICTLY CONFIDENTIAL | NOT FOR DISTRIBUTION

GREENPEAK PARTNERS

HAPPY TO HAVE YOU ON BOARD!

Prof. Stuckenschmidt and I selected 12 students for this seminar.

Wochentag	Datum	Von - Bis	Raum	
Mi	09.09.2026	17:15 - 18:45	ZOOM-Lehre-104 (Virtuelles Gebäude)	
Mi	23.09.2026	17:15 - 18:45	ZOOM-Lehre-104 (Virtuelles Gebäude)	
Mi	07.10.2026	17:15 - 18:45	ZOOM-Lehre-104 (Virtuelles Gebäude)	
Mi	21.10.2026	17:15 - 18:45	ZOOM-Lehre-104 (Virtuelles Gebäude)	
Do	05.11.2026	16:45 - 18:15	ZOOM-Lehre-104 (Virtuelles Gebäude)	→ **Today**
Di	17.11.2026	17:15 - 18:45	ZOOM-Lehre-104 (Virtuelles Gebäude)	→ **Q&A session, moved**
Mi	02.12.2026	17:15 - 18:45	ZOOM-Lehre-104 (Virtuelles Gebäude)	→ **Final group presentation**

Since we are a small group, feel free to ask questions!

GREENPEAK PARTNERS

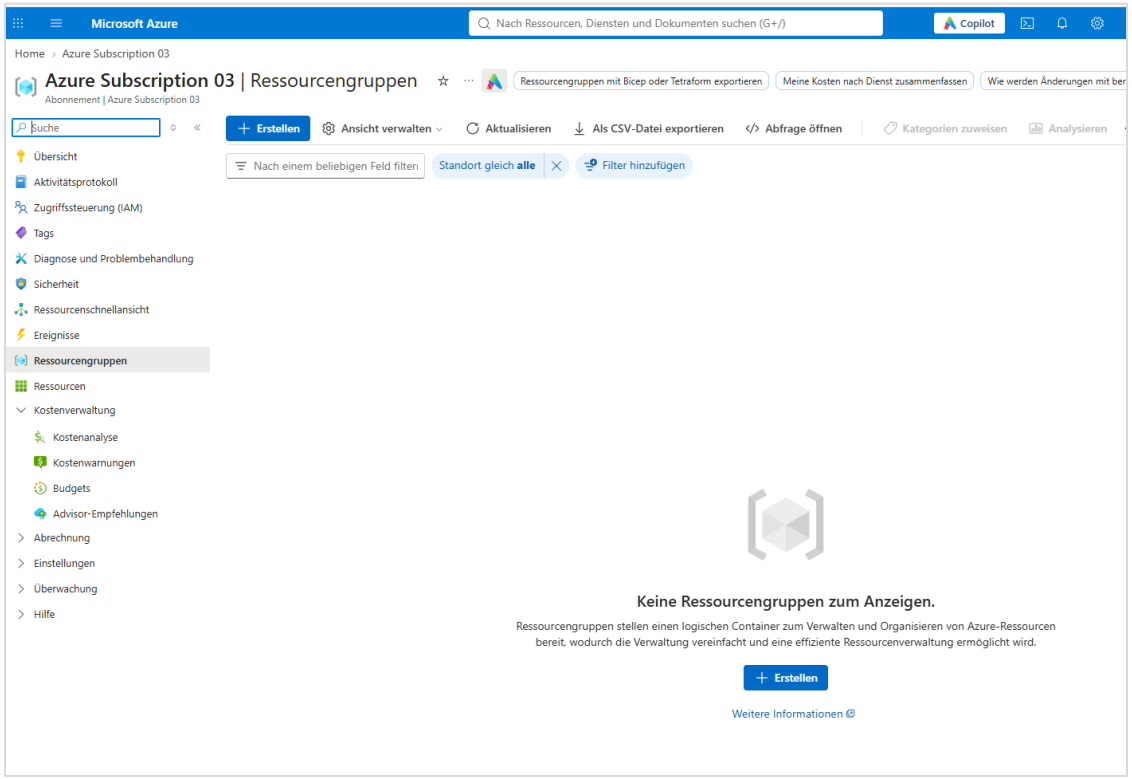
LEARNING GOALS

At the end of the seminar, you should be familiar with the following topics:

- Usage of an energy data set to predict the future electricity demand
- Basic understanding of the German electricity market and the current challenges and opportunities
- Basic understanding about what a Private Equity Company is doing and how we leverage AI in our operations
- Understand the usage of cloud technologies
- Usage of Azure to conduct a machine learning project
- Usage of Databricks to conduct cloud computation
- **Usage of Terraform to configure infrastructure as code**
- **Conduct a group work to train working in teams**

INFRASTRUCTURE AS CODE IS IMPORTANT FOR SCALABLE SOLUTIONS

Today: manual setup in the Azure portal



Manual setup works, but:

- **Difficult to debug**
- **Less transparency** on what was done
- **No change history** in GitHub
- **Slow iteration** when configurations change (e.g. regions, sizing, rights)

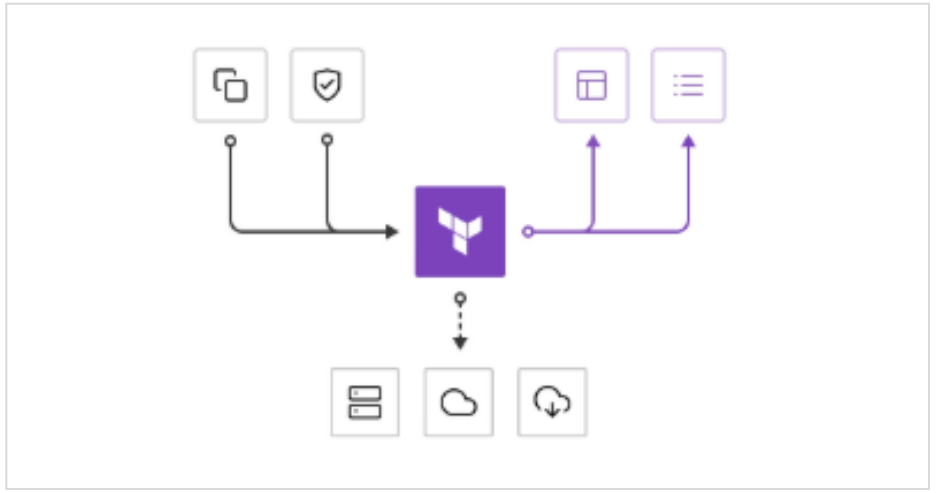
TERRAFORM IS AN INFRASTRUCTURE AS CODE TOOL

What is Terraform?

An infrastructure as code tool to build, change and version infrastructure **safely and efficiently.**

Low-level components
Compute instances, storage, networking

High-level components
DNS entries, SaaS features



Works across clouds

One tool and one configuration language for all major cloud providers

AWS **Azure** **Google Cloud**

Alternatives

- **OpenTofu** (open-source fork of Terraform)
- **AWS CloudFormation**
- **Azure ARM / Bicep**
- **Google Cloud Infrastructure Manager**

IT IS RECOMMENDED TO HAVE THE FOLLOWING STRUCTURE IN TERRAFORM

1 Core files (standard structure)

`main.tf`

Main configuration

Defines the actual resources (e.g. servers, databases, networks). For very small projects, this file alone is often enough.

`variables.tf`

Variable declarations

Declares the input variables that keep the code dynamic (e.g. region, instance size, project name): which variables exist and of what type.

`outputs.tf`

Output values

Defines what Terraform shows after deployment or passes to other modules (e.g. a server's IP address or URL).

`providers.tf`

Cloud connection

Sets which cloud providers or APIs Terraform talks to (e.g. AWS, Azure, Google Cloud, Cloudflare) and which provider versions are required. Often placed at the top of `main.tf`.

2 Environment values

`terraform.tfvars`
`*.auto.tfvars`

Sets the **actual values** for the variables declared in `variables.tf`, e.g.

```
environment = "production"
```

Often kept out of Git (may contain sensitive data)

3 Optional, one .tf file per tool

`e.g. entra.tf`

Entry Management

Regulates which groups or users can access the application

Install the Terraform CLI locally to configure your setting Terraform CLI overview | Terraform | HashiCorp Developer; it makes sense to have a separate repository only for your infrastructure code

Documentation: Terraform | HashiCorp Developer

YOU CAN CONFIGURE THE DATABRICKS SETTINGS WITH TERRAFORM SCRIPTS

Find the syntax here Databricks Terraform provider - Azure Databricks | Microsoft Learn

1 `me.tf`
Retrieves information about the current user

```
data "databricks_current_user" "me" {}
```

2 `notebook.tf`
Defines the notebook: variables, resource and URL output

```
variable "notebook_filename" { … }
resource "databricks_notebook" "this" { … }
```

3 `notebook.auto.tfvars`
Sets the notebook's properties

```
notebook_filename = "notebook-getting-started.py"
notebook_language = "PYTHON"
```

4 `notebook-getting-started.py`
Contains the notebook's code

```
display(spark.range(10))
```

5 `cluster.tf`
Defines the cluster the notebook runs on

```
data "databricks_spark_version" "latest_lts" { … }
resource "databricks_cluster" "this" { … }
```

YOU CAN START INFRASTRUCTURE SCRIPTS IN THE COMMAND LINE WITH THE AZURE SDK

Find information for the Azure SDK here Use the Azure libraries (SDK) for Python - Python on Azure | Microsoft Learn

Configure: you can use it to configure your Azure settings
Push: login locally with your terminal to push infrastructure settings directly to your Azure workspace

These are the steps to use the Azure SDK in Visual Studio Code

1 Install SDK plugin in VS Code

Shortcut: `Ctrl+Shift+X`

2 Login with az login

```
$ az login
```

3 Choose right subscription

```
$ az account set -s <id>
```

4 Redirect to your terraform folder

```
$ cd <terraform-folder>
```

5 Initialize terraform

```
$ terraform init
```

6 Apply changes

```
$ terraform apply
```

EXPERIMENT TRACKING & MODEL REGISTRY

01 Track runs

Log parameters, metrics and artifacts for each training run

02 Log the model

Save the model with its signature and environment (pip/conda)

03 Register

The registry assigns version 1, 2, 3 … linked to the run

04 Promote

Staging → Production or aliases: @challenger → @champion

What gets versioned?

- Hyperparameters and seeds
- Metrics (accuracy, RMSE …)
- Artifacts: model, plots
- Git commit and environment

Already integrated

- Databricks: managed MLflow
- Registry in Unity Catalog
- Azure ML: workspace as tracking server
- Same API → portable

In practice: stages vs. aliases

Fixed stages (Staging/Production) are deprecated in current MLflow versions. Recommended: aliases such as @champion plus tags. Unity Catalog supports aliases only.

CI/CD FOR MACHINE LEARNING

Code, data and model all change, so there are three automations: **CI** tests, **CT** trains, **CD** delivers.

01 CI

Commit / PR

Unit tests, linting, data validation (common packages uv, pytest, ruff, ty)

02 CT

Training

Databricks job or Azure ML pipeline

03 CT

Quality Gate

Proceeds only if better than @champion

04 CD

Register

New version with alias @challenger

05 CD

Deployment

Staging → approval → production

Triggers

- Code change
- New data
- Drift or schedule

Orchestration

- GitHub Actions
- Azure DevOps

IN ENTERPRISE SOLUTIONS, IT MAKES SENSE TO ORGANIZE RIGHTS IN ENTRA ID



You can configure for each application in your subscription rights. Most important ones are:

Reader

can only see the content

Contributor

can contribute, e.g. code

Owner

has privileged rights such as deletion, cost management

Overview

- Azure product to manage rights
- A developer can have more than one right
- You can organize rights in groups (E.g. owner, etc.)
- You can give rights to service accounts (e.g. Service Principals)

RISK CLASSES IN THE EU AI ACT, CENTRAL AI ACT IN THE EUROPEAN UNION

Risk class	Examples from ML practice	What follows?
Prohibited Art. 5	Social scoring, emotion recognition at work or in schools, untargeted scraping of facial images	May not be placed on the market or used
High-risk Art. 6, Annexes I and III	Applicant screening, creditworthiness, exam grading, AI in medical devices	Risk management, documentation, logging, human oversight, conformity assessment
Transparency obligation Art. 50	Chatbots, generated text and images, deepfakes	Disclose that AI is involved; label content in machine-readable form
Minimal risk All other systems	Spam filters, product recommendations, predictive maintenance	No specific obligations, except AI literacy (Art. 4)

Separate category: GPAI foundation models, with obligations for their providers regardless of the use case.

TIMELINE: WHAT APPLIES WHEN?

since 08/2024

In force

Regulation (EU) 2024/1689 enters into force

since 02/2025

Prohibitions & AI literacy

Prohibited practices (Art. 5), staff training (Art. 4)

since 08/2025

GPAI & penalties

Obligations for foundation models, national supervision and fines

since 08/2026

Transparency

Art. 50: label chatbots and AI-generated content

from 12/2027

High-risk Annex III

e.g. HR, credit, education; postponed from 08/2026

from 08/2028

High-risk Annex I

AI in regulated products, e.g. medical devices

Digital Omnibus: Regulation (EU) 2026/1744

In force since 27 July 2026. Postpones the high-risk deadlines, weakens the AI literacy obligation (“promote” instead of “ensure”) and, from 2 December 2026, adds a ban on AI for non-consensual intimate images.

Fines (Art. 99), whichever is higher

- **Prohibited practices:** up to €35M or 7% of worldwide turnover
- **Other obligations:** up to €15M or 3%
- **Incorrect information to authorities:** up to €7.5M or 1%

In companies, you will find responsible Data & AI officers

WHAT THIS MEANS FOR DAY-TO-DAY MLOPS

Documentation

Art. 11, Annex IV, Art. 18

- Purpose, design and limits of the system
- Training data: origin, quality
- Metrics and known risks
- Retention: 10 years

In the stack

- MLflow runs and registry
- Lineage in Unity Catalog
- Model card in the repository

Logging

Art. 12, 19, 26

- Automatic logs over the entire lifetime
- Make events traceable
- Retention: at least 6 months

In the stack

- Inference tables or App Insights
- Log Analytics with retention
- Model version in every log

Oversight & monitoring

Art. 14, 72, 73

- Humans can intervene and shut the system down
- Monitoring after go-live
- Report serious incidents

In the stack

- Drift and performance monitoring
- Approval gates in CI/CD
- Roles via Entra ID

Clean AI Deployment is not only important for the scalability and security of your applications; it is also important for potential audits

COMPLIANCE BY DESIGN AS A SKILL

Supervision in Germany

- **AI Implementation Act:** passed by the Bundestag in June 2026
- **Federal Network Agency (BNetzA)** as central market surveillance authority, unless a sector authority is responsible
- **KoKIVO** at the BNetzA pools expertise and coordinates
- **GDPR** continues to apply in parallel

Provider or deployer?

Whoever develops a system and makes it available under their own name is a provider. Whoever uses it within the company is a deployer. Their obligations differ considerably.

What newcomers should be able to do

01 Assess the risk class of a use case and justify it

02 Generate documentation and lineage from the pipeline instead of writing them afterwards

03 Plan logging, monitoring and approvals from the start

04 Speak a common language with legal, data protection and the business

DORA: DIGITAL OPERATIONAL RESILIENCE – SPECIAL REGULATIONS FOR FINANCIAL COMPANIES

Regulation (EU) 2022/2554 · In force since 17 Jan 2025 · Banks, insurers, payment providers · Supervisor in Germany: BaFin

ICT risk

Framework, protection, backup, recovery

Incidents

Classify serious ICT incidents and report them to the supervisor

Testing

Regular resilience tests; TLPT for large institutions

Third parties

Contracts, exit strategy, register of all ICT contracts

Info sharing

Voluntary sharing of cyber threat information

What this means for AI Ops

Azure, Databricks & co. are ICT third-party providers: contracts, exit plan and concentration risk are part of the architecture.

A model in production is an ICT system: incident process, rollback and recovery must be rehearsed.

DORA and the AI Act interlock, e.g. in a bank's credit scoring.

As a Private Equity Firm, Greenpeak is subject to increased scrutiny under the DORA Act

ANY OPEN QUESTIONS?

GREENPEAK PARTNERS

LEARNING GOALS

At the end of the seminar, you should be familiar with the following topics:

- Usage of an energy data set to predict the future electricity demand
- Basic understanding of the German electricity market and the current challenges and opportunities
- Basic understanding about what a Private Equity Company is doing and how we leverage AI in our operations
- Understand the usage of cloud technologies
- Usage of Azure to conduct a machine learning project
- Usage of Databricks to conduct cloud computation
- **Usage of Terraform to configure infrastructure as code**
- **Conduct a group work to train working in teams**

Now it’s up to you – looking forward to your solutions!


GREENPEAK PARTNERS
THANK YOU FOR
YOUR ATTENTION!
STRICTLY CONFIDENTIAL | NOT FOR DISTRIBUTION