

**Betrieb**: T-Systems Schweiz AG  
**Abteilung**: CORE Services

|  |  |  |  |
| --- | --- | --- | --- |
| **Version:** | 0.2 |  |  |
| **Berufsfachschule:** | gibb Bern |  |  |
| **Klassifizierung:** | Vertraulich |  |  |
|  |  |  |  |
| **Validexperte:** | Erich Iseli | **Fachrichtung:** | Systemtechnik |
| **Hauptexperte:** | Alex Krämer | **Projektvorgehensmodell:** | Hermes 5.1 |
| **Nebenexperte:** | Kevin Waldspurger | **Jahrgang der IPA-Durchführung:** | 2024 |
| **Verantwortliche Fachkraft:** | Mathias Hirth | **Kanton:** | Bern |
| **Berufsbildner:** | Martin Gerber | **Ausgabedatum:** | 22.04.2024 |

Betrieb: T-Systems Schweiz AG  
Abteilung: CORE Services

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Version 0.0 (Status)

vom 22.04.2024

Wählen Sie einen Geheimhaltungsvermerk aus

Änderungshistorie

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Stand | Bearbeiter\*in | Änderungen/Kommentar |
| 0.1 | 22.04.2024 | Justin Winistörfer | Schreiben der Kapitel: Aufgabenstellung, Standards, IPA-Schutzbedarfsanalyse und Organisation der IPA Ergebnisse |
| 0.2 | 23.04.2024 | Justin Winistörfer | Schreibe der Kapitel: Projektvorgehen, Projektorganisation und Projektrollen, Einleitung, Ist- Soll Analyse und einfügen eines Netzplanes. |
| 0.3 | 24.04.2024 | Justin Winistörfer | Schreiben der Kapitel: Ziele, Anforderungen, System-, Namens-, Netzwerk-, ISDS- und Testkonzept. Allgemeine kleine Rechtschreibkorrekturen. |
| 0.4 | 26.04.2024 | Justin Winistörfer | Schreiben der Kapitel: Testkonzept, Automatisierungskonzept, Phasenfreigabe, IDE Umgebung, Zugang zu AWS und Terraform |

Kurzfassung des IPA Berichtes

Inhaltsverzeichnis

[Teil 1: Formaler Teil 1 7](#_Toc164845148)

[1 Aufgabenstellung 7](#_Toc164845149)

[1.1 Titel der Arbeit 7](#_Toc164845150)

[1.2 Ausgangslage 7](#_Toc164845151)

[1.3 Detaillierte Aufgabenstellung 7](#_Toc164845152)

[1.3.1 Erwartete Resultate des Auftraggebers 7](#_Toc164845153)

[1.3.2 Prüfbare/Messbare Ziele 8](#_Toc164845154)

[1.3.3 Zu erstellende Dokumentation 9](#_Toc164845155)

[1.3.4 Meilensteine 9](#_Toc164845156)

[1.3.5 Funktionale Anforderungen 9](#_Toc164845157)

[1.3.6 Nicht-funktionale Anforderungen 10](#_Toc164845158)

[1.4 Mittel und Methoden 10](#_Toc164845159)

[1.4.1 Mittel 10](#_Toc164845160)

[1.4.2 Methoden 10](#_Toc164845161)

[1.5 Vorkenntnisse 10](#_Toc164845162)

[1.6 Vorarbeiten 11](#_Toc164845163)

[1.7 Neue Lerninhalte 11](#_Toc164845164)

[1.8 Arbeiten in den letzten 6 Monaten 11](#_Toc164845165)

[2 Standards 12](#_Toc164845166)

[2.1 Dokument 12](#_Toc164845167)

[2.2 Firmenexterne Standards 12](#_Toc164845168)

[3 IPA-Schutzbedarfsanalyse 12](#_Toc164845169)

[4 Organisation der IPA Ergebnisse 12](#_Toc164845170)

[4.1 Arbeitsplatz 12](#_Toc164845171)

[4.2 Dokumentenablage 13](#_Toc164845172)

[4.3 Versionierung 13](#_Toc164845173)

[4.4 Dokumentensicherung 14](#_Toc164845174)

[4.5 Wiederherstellung 14](#_Toc164845175)

[4.6 Test der Wiederherstellung 15](#_Toc164845176)

[5 Projektvorgehen 16](#_Toc164845177)

[5.1 Projektmethode 16](#_Toc164845178)

[5.2 Phasen 16](#_Toc164845179)

[5.3 Szenario 16](#_Toc164845180)

[5.4 Module 17](#_Toc164845181)

[5.5 Meilensteine 18](#_Toc164845182)

[6 Projektorganisation 19](#_Toc164845183)

[6.1 Projektorganisation 19](#_Toc164845184)

[6.2 Projektrollen 20](#_Toc164845185)

[6.3 Kontaktangaben 20](#_Toc164845186)

[7 Zeitplan 21](#_Toc164845187)

[8 Arbeitsjournale 21](#_Toc164845188)

[8.1 Arbeitsjournal vom 22.04.2024 21](#_Toc164845189)

[8.2 Arbeitsjournal vom 23.04.2024 23](#_Toc164845190)

[Teil 2: Individueller Praktischer Teil 26](#_Toc164845191)

[9 Einleitung 26](#_Toc164845192)

[9.1 T-Systems Schweiz AG 26](#_Toc164845193)

[9.2 Core SERVICES 26](#_Toc164845194)

[9.3 Begründung für die IPA-Themen Wahl 26](#_Toc164845195)

[10 Initialisierung 28](#_Toc164845196)

[10.1 Ist- / Soll-Analyse 28](#_Toc164845197)

[10.1.1 Ist-Zustand 28](#_Toc164845198)

[10.1.2 Soll-Zustand 31](#_Toc164845199)

[10.2 Ziele 32](#_Toc164845200)

[10.2.1 SMART 32](#_Toc164845201)

[10.2.2 Persönliche Vorgehensziele 32](#_Toc164845202)

[10.2.3 Projektziele 33](#_Toc164845203)

[10.3 Anforderungen 33](#_Toc164845204)

[10.4 Variantenentscheid 34](#_Toc164845205)

[10.5 Projektfreigabe 34](#_Toc164845206)

[Abbildungsverzeichnis 35](#_Toc164845207)

[Tabellenverzeichnis 36](#_Toc164845208)

[11 Einleitung 36](#_Toc164845209)

[11.1 Dokument als Standardvorlage nutzen 36](#_Toc164845210)

[11.2 Übersicht über die wichtigsten Textformatierungen 36](#_Toc164845211)

[11.3 Formatierung von Texten 37](#_Toc164845212)

[12 Der Umgang mit Bildern und Tabellen 38](#_Toc164845213)

[12.1 Platzierung von Bildern 38](#_Toc164845214)

[12.2 Platzierung und Formatierung von Tabellen 39](#_Toc164845215)

[13 Aufzählungen 40](#_Toc164845216)

[13.1 Punktaufzählungen 40](#_Toc164845217)

[13.2 Aufzählungen mit Nummerierung 41](#_Toc164845218)

[13.2.1 Neubeginn einer Nummerierung 42](#_Toc164845219)

[14 Beispielanhang 43](#_Toc164845220)

[Mitgeltende Unterlagen 44](#_Toc164845221)

[Abkürzungsverzeichnis/Glossar 45](#_Toc164845222)

[Quellenverzeichnis 47](#_Toc164845223)

# Teil 1: Formaler Teil 1

# Aufgabenstellung

## Titel der Arbeit

Chaos-Engineering mit Chaos-Mesh

## Ausgangslage

Wir von T-Systems sind auf Cloud-Infrastrukturen und Mikroservices spezialisiert. Zudem sind wir gerade im Aufbau unserer eigenen Cloud Infrastruktur. In der Vergangenheit gab es mehrere Vorfälle, bei denen unerwartete Systemausfälle zu erheblichen Unterbrechungen der Dienstleistungen führten. Damit die Zuverlässigkeit und Ausfallsicherheit in Zukunft weiter verbessert werden kann, möchten wir das Tool "Chaos-Mesh" in Betrieb nehmen, um diese in Zukunft für unsere Infrastruktur als Test-Tool verwenden zu können. Damit die Chaos Experimente angewendet werden können, wird im Voraus der IPA eine Full-Stack Applikation bereitgestellt, welche ein Kinoticketverkauf System simuliert. Es gibt verschiedene Filme, für die es die Möglichkeit gibt, Tickets zu reservieren und zu kaufen. Anhand diesem Use-Case müssen verschiedene Experimente durchgeführt und der Einfluss auf den Kunden gemessen werden.

## Detaillierte Aufgabenstellung

### Erwartete Resultate des Auftraggebers

Es müssen die notwendigen Konzepte erstellt werden. Sie bilden die Grundlage für die Realisierung des Systems:

* Namenskonzept
* Systemkonzept
* Netzwerk-/Portkonzept
* Berechtigungskonzept
* Testkonzept
* ISDS-Konzept

Die notwendige Infrastruktur muss mit Terraform realisiert werden. Dies umfasst die Bereitstellung der erforderlichen Netzwerke (VPC, Subnets), Internet Gateway, NAT, Route Tables, IAM-Rollen sowie des EKS-Clusters. Ein kritischer Aspekt dabei ist, den Terraform-Zustand als „Remote Storage“ in einem S3-Bucket zu speichern. Der so entstandene Code muss in GitHub mit Versionskontrolle hinterlegt werden. Zudem ist das Erstellen einer Pipeline erforderlich, um die Kino-App automatisiert deployen zu können.  
  
Auf der eingerichteten Infrastruktur muss Chaos-Mesh installiert und konfiguriert werden. Es muss sich dabei strikt an die Dokumentation von <https://chaos-mesh.org/docs/production-installationusing-> helm/ gehalten werden. Die folgenden Experimente müssen an der Kino App durchgeführt werden:

* Netzwerkchaos: Simulation von Netzwerkverzögerungen, Paketverlusten und instabilen Netzwerkverbindungen.
* Podchaos: Simulation von Pod-Ausfällen und Überprüfung der Selbstheilungsfunktionen.
* CPU- und Speicherchaos: Künstliche Überlastung der CPU oder der Speicherressourcen.
* HTTP-Chaos: Manipulation von HTTP-Anfragen und -Antworten, einschliesslich Verbindungsabbruch, Einführung von Verzögerungen, Inhaltsersetzung oder -ergänzung.
* Service-Abhängigkeiten: Simulation von Ausfällen in Abhängigkeiten (wie Datenbanken), um die Auswirkungen auf die Anwendung zu beobachten.

### Prüfbare/Messbare Ziele

Infrastruktur mit Terraform:

* Bereitstellung der vollständigen Infrastruktur.
* Implementierung und Dokumentation von VPC, Subnets, Internet Gateway, NAT, Route Tables,
* IAM-Rollen und EKS-Cluster unter Verwendung von Terraform.
* Sicherstellung, dass der Terraform-Zustand sicher in einem S3-Bucket als „Remote Storage“
* gespeichert ist.

Automatisierte Deployment-Pipeline:

* Erstellung und Implementierung der CI/CD-Pipeline in "Github Actions" die automatisches Deployment der Kino-App ermöglicht.
* Podchaos muss mindestens in die Deployment-Pipeline integriert werden, um die Anwendungsmöglichkeiten von Chaos-Mesh in automatisierten Deployments zu demonstrieren.

Installation und Konfiguration von Chaos-Mesh:

* Erfolgreiche Installation und Konfiguration von Chaos-Mesh auf der eingerichteten Infrastruktur.
* Durchführung aller geplanten Experimente (Netzwerkchaos, Podchaos, CPU- und Speicherchaos, HTTP-Chaos, Service-Abhängigkeiten).

GitHub-Versionierung:

* Jeglicher Code und Dokumentation müssen in Github versioniert werden. Es muss mindestens einen tagesaktuellen commit geben. Ein neues Repository für die IPA muss erstellt werden.

YAML-Codierung:

* Implementierung der Chaos-Experimente mittels YAML als wiederverwendbarer Code

Chaos Engineering Experimente:

* Die Auswirkung der Chaos-Experimente auf die Kino App müssen analysiert werden mit Hilfe von CloudWatch. Dabei ist es wichtig, dass erkannt wird, welche Metriken sich durch die Chaos- Experimente verändern (z.B. Latenz). Dies ist als Beispiel in der Anleitung für das Fachteam zu dokumentieren.

Dokumentation und Anleitung:

* Erstellung einer umfassenden, schrittweisen Anleitung zur Implementierung und Nutzung von Chaos-Mesh in Projekten zur Verbesserung der Systemresilienz.

### Zu erstellende Dokumentation

* Benutzerhandbuch: Anleitung zur Verwendung von Chaos Mesh fürs technische Team

### Meilensteine

* Erfolgreiche provisionierung der Infrastruktur
* Erfolgreiche Installation von Chaos-Mesh
* Erfolgreiches erstellen der gennanten Experimente
* Erfolgreiche Durchführung der Experimente
* Erfolgreiche analyse Auswirkung auf die Kino App
* Erfolgreiche Implementierung eines Tests in der Pipeline
* Dokumentation der Ergebnisse
* Abschluss des IPA Berichts (Technische Dokumentation, Benutzerhandbuch etc.)

### Funktionale Anforderungen

* Bereitstellung der notwendigen Infrastruktur mit Terraform: Automatisierung der Infrastrukturerstellung, einschliesslich Netzwerke (VPC, Subnets), Internet Gateway, NAT, Route Tables, IAM-Rollen, und des EKS-Clusters.
* Implementierung von Chaos-Experimenten: Durchführung von spezifizierten Chaos-Experimenten (Netzwerkchaos, Podchaos, CPU- und Speicherchaos, HTTP-Chaos, Service-Abhängigkeiten) an der Kino-App.
* Analyse der Auswirkungen von Chaos-Experimenten: Nutzung von CloudWatch zur Überwachung und Analyse der durch die Chaos-Experimente veränderten Metriken (z.B. Latenz).
* Integration eines Chaos-Experiments in die Deployment-Pipeline: Automatisierung der Einbindung mindestens eines Chaos-Experiments in den Deployment-Prozess der Kino-App.
* Erstellung einer Anleitung: Entwicklung einer detaillierten Anleitung zur Implementierung von Chaos-Mesh in zukünftigen Projekten und zur kontinuierlichen Verbesserung der Systemresilienz.

### Nicht-funktionale Anforderungen

* Verwendung von Remote Storage für Terraform-Zustand: Speicherung des Terraform-Zustands in einem S3-Bucket.
* Versionskontrolle: Speicherung des gesamten Codes, inklusive Terraform und YAML für Chaos- Experimente, in GitHub.
* Automatisiertes Deployment: Entwicklung einer Pipeline, die ein automatisiertes Deployen der Kino-App ermöglicht.
* Wiederverwendbarkeit von Code: Implementierung der Chaos-Experimente als wiederverwendbaren Code mittels YAML.
* Dokumentation: Bereitstellung einer klaren, schrittweisen Anleitung für das Fachteam, einschliesslich Beispielen, wie die Auswirkungen der Chaos-Experimente erkannt und analysiert werden können.

## Mittel und Methoden

### Mittel

* "Chaos Mesh" als das Hauptwerkzeug für Chaos-Experimente.
* "CloudWatch" Zur Überwachung und Analyse der Systemleistung und -stabilität.
* "Github" als Versionierungs Tool
* "Github Actions" als Pipeline Tool
* "S3 Bucket" zum speichern des Terraform States
* "Terraform" zum provisionieren von Infrastruktur
* VSCode
* BYOD
* Office 365
* AWS
* Linux
* Helm
* Terraform
* EKS (Kubernetes)
* Firmenstandard für Dokumentationen (Layout, Fonts, Formatierung)
* Helm

### Methoden

* GitOps
* CICD
* HERMES Projektstrukturierung: Anwendung der HERMES-Methodik für die Projektstrukturierung, inklusive klarer Rollen, Verantwortlichkeiten und Prozesse.

## Vorkenntnisse

AWS (1 Jahr)

Grundlagen Kubernetes (0,5 Jahr)

Netzwerkkenntnisse (3,5 Jahre)

Docker (1 Jahr)

Full-Stack Applikationsarchitektur (1,5 Jahre)

Linux (4 Jahre)

Terraform (2 Jahre)

## Vorarbeiten

* Testaccount auf AWS besorgen
* Entwicklung der Kino App
* IPA-Berichtsvorlage

Grundlagen Chaos-Engineering

* Verständnis von Chaos Engineering: Lernen, was Chaos Engineering ist, warum es wichtig ist und wie es zur Verbesserung der Systemresilienz beiträgt
* Prinzipien und Methodik: Verständnis der Grundprinzipien und Methoden des Chaos Engineering, einschliesslich der Planung und Durchführung von Chaos-Experimenten.

## Neue Lerninhalte

* Chaos Mesh Grundlagen: Erlernen der Installation, Konfiguration und Nutzung von Chaos Mesh.
* Experimente mit Chaos Mesh: Verständnis, wie man verschiedene Arten von Experimenten (wie Netzwerklatenz, Ressourcenknappheit, etc.) mit Chaos Mesh durchführt.
* Systemüberwachung: Lernen, wie man Systeme überwacht, um Leistungsindikatoren und potenzielle Probleme zu erkennen.
* Fehleranalyse: Fähigkeiten zur Identifizierung, Analyse und Behebung von Systemfehlern und ausfällen.
* Datenanalyse und Berichterstattung: Lernen, wie man aus Experimenten gewonnene Daten analysiert und Ergebnisse effektiv berichtet.

Welche Quellen stehen ihm zur Verfügung?

ChatGPT, Google

## Arbeiten in den letzten 6 Monaten

Der Lernende hatte sich grossteils auf die SwissSkills vorbereitet (AWS). Zudem hat er für einen Kunden mehrere Änderungen an einem Angular Frontend durchgeführt welches mit Typescript

geschrieben ist.

# Standards

## Dokument

Alle gängige Textdokumente, welche innerhalb der T-Systems erstellt werden, müssen die Standards der vorgegebenen Dokumentenvorlage einhalten. Ebenso zur Erstellung dieser IPA Dokumentation wurde die firmeninterne Dokumentenvorlage eingesetzt und verwendet.

## Firmenexterne Standards

Aufgrunde dessen, dass diese IPA komplett ausserhalb des firmeninternen Bereiches geschieht, sind keine weiteren Standards anzuwenden. Standards welche während der IPA eingesetzt werden, werden in der Konzeptphase dieses Projektes definiert und angewendet.

# IPA-Schutzbedarfsanalyse

Dieses Dokument erhält die Klassifizierung „Vertraulich“, da während der Dokumentation sensible Daten und Informationen ersichtlich sind, wie: IP-Adressen, Namen sowie E-Mail-Adressen, welche nicht für die Öffentlichkeit bestimmt sind. COMMENT: Eventuell ergänzen

# Organisation der IPA Ergebnisse

Nachfolgend wird die Organisation des Arbeitsplatzes, sowie die Dokumentenverwaltung, beschrieben.

## Arbeitsplatz

Die IPA wird im Hauptsitz der T-Systems Schweiz, in Zollikofen, durchgeführt. Für die IPA wurde ein Co-Work Raum, für die Zeitspanne während der IPA, reserviert, damit ungestört gearbeitet werden kann. Der Arbeitsplatz umfasst folgende Sachen: Ein höhenverstellbarer Tisch, ein Bürostuhl, ein Monitor und das eigene Arbeitsgerät. Ebenso wird der Schreibtisch verwendet, um ausgedruckte Dokumente stets Griffbereit zu haben. Der Arbeitsplatz sieht wie folgt aus:  
  
Ein Bild, das Im Haus, Mobiliar, Computermonitor, Computer enthält.

Automatisch generierte Beschreibung

Abbildung 1: Arbeitsplatz während der IPA

## Dokumentenablage

Alle Dokumente, welche während der IPA verwendet und erstellt werden, sind auf dem eigenen OneDrive Account gesichert, nachfolgend ist der Pfad für den Speicherort aufgelistet: Eigene Dateien > IPA > Dokumente.

## Versionierung

Die Versionierung aller Dokumente, welche während der IPA erstellt werden, einschliesslich dieses Dokumentes, werden in Tagesblöcken strukturiert. Die Dokumentation wird täglich auf der vorher erwähnten Dokumentenablage gesichert. Dabei wird in der Änderungshistorie jeweils die tagesaktuelle Version eingesetzt, sowie ein Kommentar. Sichert man nun das Dokument am Ende des Tages in der Dokumentenablage, erstellt OneDrive automatisch eine eigene Versionierung, welche wie folgt aussieht:  
  
Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 2: Versionierungsverlauf in Microsoft OneDrive

Da die Versionsnummer im Versionsverlauf von OneDrive, nicht mit dem selbstdefinierten Versionierungsschema übereinstimmt, ist es umso wichtiger, dass eine konsistente und detaillierte Änderungshistorie im Dokument vorhanden ist.

COMMENT: Müssen die Scripts hier auch erwähnt werden?

## Dokumentensicherung

Die IPA Dokumentation wird auf zwei verschiedenen Arten gesichert, damit ein Verlust der Dokumentation vermieden werden kann.

1. OneDrive  
   OneDrive ist das Hauptmedium während der IPA, auf welchem die Dokumentation gesichert wird.
2. GitHub  
   Nebst allen Scripts, welche auf GitHub gesichert werden, wird ebenso auch die Dokumentation im selben Repository hochgeladen. Dabei wird jeden Abend ein Commit erstellt, in welchem, in einer Commit-Nachricht, die aktuelle Dokumentationsversion eingefügt wird. So kann man im Verionsverlauf von GitHub die einzelnen Commits rückverfolgen und kann so den „Dokumentationscommit“ von den anderen Commits unterscheiden.  
     
   Ein Bild, das Text, Screenshot enthält.

   Automatisch generierte Beschreibung

Abbildung 3: Versionierungsverlauf in GitHub

## Wiederherstellung

In beiden Speichermedien für die IPA Dokumentation, ist eine automatische Versionierung bereits eingebaut. Duch diese Versionierung, kann man stets auf eine vorherige Version zugreifen und diese Version herunterladen. Diese Versionierung bleibt auch dann bestanden, falls ein Dokument versehentlich gelöscht worden sein soll.

## Test der Wiederherstellung

Nachfolgend wird zwischenzeitlich ein Dummy-Versioneriungseintrag vorgenommen, und auf beiden Plattformen erneut hochgeladen, um die Funktionalität der Versionierung, auf beiden Plattformen zu überprüfen. Nachfolgend sieht man von beiden Plattformen den Versionsverlauf, man beachte bei OneDrive zudem die veränderte Dokumentengrösse, welche auf eine Änderung hinweist:  
  
Microsoft OneDrive:  
Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Abbildung 4: Aktualisierter Versionsverlauf in Microsoft OneDrive

GitHub:  
Ein Bild, das Text, Screenshot enthält.

Automatisch generierte Beschreibung

Abbildung 5: Aktualisierter Versionsverlauf in GitHub

Bei beiden Versionsverläufen, kann stets eine vorherige Version heruntergeladen werden. Dies bestätigt, dass eine Wiederherstellung der Dokumentation jederzeit möglich ist.

# Projektvorgehen

## Projektmethode

Um dieses IPA Projekt umzusetzen, wurde als passende Porjektmanagementmethode HERMES 5.1 ausgewählt. Folgend ist eine Definition von HERMES einzusehen:  
  
„HERMES ist die Projektmanagementmethode für Projekte im Bereich der Informatik, der Entwicklung von Dienstleistungen und Produkten sowie der Anpassung der Geschäftsorganisation. HERMES unterstützt die Steuerung, Führung und Ausführung von Projekten verschiedener Charakteristiken und Komplexität. HERMES hat eine klare, einfach verständliche Methodenstruktur, ist modular aufgebaut und erweiterbar.“[[1]](#footnote-1)

## Phasen

Dadurch, dass sich HERMES als Wasserfallmodell auszeichnet, wird die Projektstruktur in vier aufeinanderfolgende Phasen aufgebaut, diese sind bei HERMES wie folgt gegliedert:



Abbildung 6: Phasenablauf HERMES

## Szenario

HERMES unterscheidet verschiedene Projektaufgaben in sogenannten Szenarien. Dabei gibt es bereits schon vorgefertigte Szenarien, aber es können auch eigene Szenarien erstellt werden. Ein Szenario enthält dabei einen Strukturplan, welcher die verschiedenen Module und Meilensteine aufzeigt, welche verwendet werden sollten. Während diesem Projekt, wird ein eigenes Szenario, basierend auf dem „IT-Standardanwendung“ Szenario verwendet, da in diesem Projekt lediglich der Test der Software im Fokus stehe und nicht die Integrierung in den produktiven und organisatorischen Betrieb. Im nachfolgenden Unterkapitel wird aufgeführt, welche Module dabei verwendet oder ausgegrenzt werden.[[2]](#footnote-2)

## Module

Für das individuelle Szenario, welches in diesem Projekt angewendet wird, wurden folgende Module ausgewählt, welche diverse Aufgaben enthalten und somit die Projektführung unterstützen:

Ein Bild, das Text, Screenshot, Visitenkarte, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 7: Hermes Module

Nachfolgend wird tabellarisch aufgelistet, welche Inhalte die jeweiligen Module enthalten:[[3]](#footnote-3)

|  |  |
| --- | --- |
| Module | Beschreibung |
| Projetsteuerung | Das Projekt wird initialisiert und anschliessend eine kontinuierliche Steuerung des Projektes durch Ziele, Anforderungen und Vorgaben. |
| Projektführung | Durchführung und Planung des Projektes. Einhaltung der definierten Rahmenbedingungen und Anforderungen. |
| Projektgrundlagen | Als Grundlage wird eine IST-/SOLL-Analyse erstellt, welche als Grundlage für die Weiterführung des Projektes dient. |
| IT-System | Ein IT-System wird konzipiert, umgesetzt und dokumentiert. Relevante Anforderungen werden berücksichtigt. |
| Testen | Tests werden konzipiert, vorbereitet, durchgeführt und dokumentiert. |
| Informationssicherheit und Datenschutz | Informationssicherheit und Datenschutzanforderungen werden ermittelt, konzipiert und umgesetzt. |

COMMENT: Tabellenverzeichnis

## Meilensteine

Meilensteine sind von grosser Bedeutung in der Projektmethode HERMES. Jede Phase, welche in HERMES durchlaufen wird, erhält vor oder nach jeder Phase einen Meilenstein, welcher die Qualitätssicherung während des Projektes sicherstellen soll. Zudem werden während einer Phase, wichtige Ereignisse, zusätzlich als Meilenstein eingebaut. Wird während der Projektumsetzung ein Meilenstein erreicht, so wird die Qualität geprüft und danach entschieden, ob das Projekt weitergeführt werden kann, oder ob zuerst noch die Qualität sichergestellt werden muss.[[4]](#footnote-4) In diesem IPA-Projekt sind die Meilensteine wie folgt definiert:  
  
  
Ein Bild, das Text, Screenshot, Schrift, Grafiken enthält.

Automatisch generierte Beschreibung

Abbildung 8: Meilensteine HERMES

Die aufgelisteten Meilensteine wurden so erstellt, dass diese auf das eigene individuelle Szenario abgestummen sind. So wurde beispielsweise der Meilenstein „Variantenentscheid“ entfernt, da bereits in diesem Projekt die Software schon vorgegeben ist. In der untenstehenden Tabelle werden die Meilensteine genauer erläutert:

|  |  |  |
| --- | --- | --- |
| ID | Meilensteine | Beschreibung |
| M1 | Projektinitialisierungsauftrag | Dieser Meilenstein ist erfüllt, sobald der Detailbeschrieb validiert wurde. Anschliessend kann das Projekt termingerecht mit der Initialisierungsphase gestartet werden. |
| M2 | Projektfreigabe | Nach Abschluss der Initialisierung, wird die Konzeptphase freigegeben. |
| M3 | Testkonzept | Sobald ein Testkonzept erstellt wurde, zählt dieser Meilenstein als erreicht. |
| M4 | Phasenfreigabe | Nach Abschluss der Konzeptphase, wird die Realisierungsphase freigegeben. |
| M5 | Provisionierung der Infrastruktur | Sobald die Provisionierung der Infrastruktur durch Terraform Scripts realisiert wurde, zählt dieser Meilenstein als erreicht. |
| M6 | Automatisierung eines Chaos-Experimentes | Sobald eine Automatisierung eines beliebigen Chaos-Experimentes stattgefunden hat, zählt dieser Meilenstein als erreicht. |
| M7 | Phasenfreigabe 2 | Nach Abschluss der Realisierungsphase, wird die Einführungsphase freigegeben. |
| M8 | Projektabschluss | Dieser Meilenstein zählt als erreicht, sobald das Projekt abgeschlossen wurde. |

COMMENT: Tabellenverzeichnis

# Projektorganisation

## Projektorganisation

Ein Bild, das Text, Visitenkarte, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 9: Projektorganisation

## Projektrollen[[5]](#footnote-5)

|  |  |
| --- | --- |
| Rollen | Beschreibung der Rolle |
| Auftraggeber | Der Auftraggeber ist verantwortlich für die Ergebnisse des Projekts und die Erreichung der Ziele. |
| Projektausschuss | Der Projektausschuss ist verantwortlich für die Unterstützung des Auftraggebers und vertritt Interessen des Unternehmens. |
| Berufsbildner | Der Berufsbildner ist die verantwortliche Fachperson für den Lehrling. |
| Qualitäts- und Sicherheitsmanagement | Das Qualitäts- und Sicherheitsmanagement ist verantwortlich für die Bewertung des Projektes unabhängig des Kriterienkatalogs. |
| Projektleitung | Der Projektleiter ist der Ansprechspartner für alle Beteiligten innerhalb des Projekts. Seine Aufgabe umfasst das leiten des Projekts. |
| Projektmitarbeiter | Der Projektmitarbeiter setzt die Anforderungen und Ziele um. |
| Fachspezialist | Der Fachspezialist dient als Anlaufsstelle für Fragen bei technischen Schwierigkeiten im Projekt. |
| Tester | Der Tester führt die Tests durch, welche innerhalb der Konzeptphase vordefiniert wurden. Alle Tests müssen vom Tester dokumentiert werden. |

## Kontaktangaben[[6]](#footnote-6)

|  |  |
| --- | --- |
| Person | Kontaktdaten |
| Justin Winistörfer | E-Mail: [Justin.Winistoerfer@t-systems.com](mailto:Justin.Winistoerfer@t-systems.com) Mobile: +41 76 681 75 63 |
| Mathias Hirth | E-Mail: [Mathias.Hirth@t-systems.com](mailto:Mathias.Hirth@t-systems.com)  Mobile: +41 76 382 71 39 |
| Martin Gerber | E-Mail: [Martin.Gerber2@t-systems.com](mailto:Martin.Gerber2@t-systems.com) Mobile: +41 78 784 25 47 |
| Alex Krämer | E-Mail: [Alex.Kraemer@wagner.ch](mailto:Alex.Kraemer@wagner.ch)  Mobile: +41 79 697 12 76 |
| Kevin Waldspurger | E-Mail: [Kevin.Waldspurger@besonet.ch](mailto:Kevin.Waldspurger@besonet.ch)  Mobile: +41 79 712 48 49 |
| Erich Iseli | E-Mail: Mobile: |

# Zeitplan

# Arbeitsjournale

## Arbeitsjournal vom 22.04.2024

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte Personen | SOLL Zeit (h) | IST Zeit (h) |
| Zeitplan erstellen | Justin Winistörfer | 4.5 | 4.5 |
| Aufgabenstellung integrieren | Justin Winistörfer | 0.75 | 1 |
| Firmenstandards | Justin Winistörfer | 0.5 | 0.25 |
| IPA-Schutzbedarfsanalyse | Justin Winistörfer | 0.75 | 0.5 |
| Organisation der IPA Ergebnisse | Justin Winistörfer | 1 | 1.5 |
| Arbeitsjournal | Justin Winistörfer | 0.5 | 0.5 |
| Gesamtzeit | - | 8 | 8.25 |

**Tagesablauf:**

Heute war nach etwas warten endlich der Start meiner IPA. Um die nächsten 10 Tage, der heutige Tag inkludiert, möglichst genau planen kann und auch nicht in einen Zeitstresse kommen kann und da es sowieso eine Pflicht ist, habe ich als allererstes begonnen meinen Zeitplan zu erstellen. Dafür habe ich intern eine Excel Vorlage genommen, in welchen ich meine nötigen Sachen eintragen kann. Somit musste ich mir erstmal Gedanken machen, wie ich einen Zeitplan aufbaue und besonders welche Tätigkeiten alle in meine IPA vorkommen. Sobald ich mir die Tätigkeiten eingetragen habe, musste ich mir überlegen, welche Task wieviel Zeit in Anspruch nimmt. Es war zuerst recht schwer alles einzuschätzen, da ich zuvor noch nie wirklich einen Zeitplan erstellt habe. Also begann ich mal die Zeiten in die Tätigkeiten einzutragen. Sobald ich alles eintragen habe, habe ich bemerkt, dass ich bei diversen Tasks mehr Zeit brauche und dann musste ich überall die Felder wie verschieben und basteln bis es passt. Ebenso musste ich mich auch noch informieren, welche Aufgaben in welche HERMES Phase gehören was ebenso viel Zeit beansprucht. Es war definitiv ein sehr langwirriger Task und hat mich viel mehr Zeit gekostet als ich dachte. Zum Glück muss der Zeitplan nur einmalig erstellt werden.

Nachdem der Zeitplan erstellt war, habe ich mir die firmeninternen Dokumentenvorlage zu Hand genommen und begonnen die Aufgabenstellung ins Dokument zu integrieren. Leider war auch dieser Task etwas mühsam, da bei kopierten Textinhalten bei Zeilenunterbrüchen ein neuer Absatz ins Dokument eingefügt wurde. Somit musste ich diese Absätze immer wieder herauslöschen, was mich anschlissend mehr Zeit gekostet hat, als ich zuerst geplant habe. Bereits der erste Zeitübertritt, ups.

Danach habe ich ein Kapitel über unsere Firmenstandards geschrieben, da wir keine wirklichen Vorgaben, besonders bei meiner IPA, haben, war das Textchen dazu schnell geschrieben.

Die Organisation der IPA Ergebnisse hatte mich jedoch auch wieder mehr Zeit gekostet als geplant, da ich noch die Versionierung & Wiederherstellung testen musste.

Jetzt sitze ich noch hier und schreibe mein Arbeitsjournal. Wie lange mir das wohl freude bereiten würde.

**Hilfestellungen:**

Mit der HERMES Dokumentation von <https://hermes.admin.ch/> habe ich mir eine kurze Übersicht verschafft, welche Tätigkeiten in welche Phasen gehören.

**Reflexion:**

Das lief gut:

Die jeweiligen Kapitel waren recht schnell umgesetzt, da ich bereits schon alle nötigen Informationen vom Betrieb kannte und die jeweiligen Kapitel auch sehr selbsterklärend sind. Durch das konnte ich mir Zeit sparen um Beispielsweise zuerst nachzufragen, ob es eine Dokumentenvorlage in unserem Betrieb gäbe oder nicht.

Das lief nicht so gut:  
Ich hatte zuerst sehr grosse Mühe einen Zeitplan zu erstellen, da ich zuvor noch nie einen Zeitplan erstellt habe und ebenso auch genau überlegen musste, welche Aufgaben während der IPA vorkommen. Da mir nicht die ganze IPA bekannt ist, musst ich auch einen Puffer für unbekannte Aufgaben einbauen. Es hat zuerst recht lange gedauert, bis ich mich einmal in den Zeitplan eingefunden habe.   
  
Meine heutigen Erkenntnisse:  
Ich denke das die Wahl von Excel eventuell nicht als geeignetes Tool scheint um einen Zeitplan zu erstellen, jedoch habe ich mich auch nicht informiert ob es Alternativen gäbe, da ich einfach nur schleunigst starten wollte und somit die Firmeninternevorlage genommen habe, welche bereits schon von vorherigen Lehrlingen erstellt wurde. Für einen künftigen Zeitplan würde ich Excel nicht mehr verwenden.  
  
**Meine Schlussbilanz:**Ich denke, dass ich heute einen guten Start abgelegt habe und denke, dass ich die nächsten 10 Tage mit bravur überstehen werde und danach hoffentlich noch freude an Excel habe.

**Nächste Schritte:**Da ich für den heutigen Tag, im Zeitplan noch dabei bin, kann ich morgen nach Zeitplan weiterarbeiten und mit dem Kapitel "Projektvorgehen" beginnen. Ebenso sollte ich mir auch Fragen aufschreiben, welche ich beim morgigen Expertenbesuch stellen kann.

## Arbeitsjournal vom 23.04.2024

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte Personen | SOLL Zeit (h) | IST Zeit (h) |
| Projektvorgehen | Justin Winistörfer | 1 | 1.50 |
| Projektorganisation und Projektrollen | Justin Winistörfer | 0.75 | 0.50 |
| Teil 2: Einleitung | Justin Winistörfer | 1 | 1 |
| Reserve Formaler Teil 1 | Justin Winistörfer | 0.50 | 0.50 |
| Abschluss Formaler Teil 1 | Justin Winistörfer | 0.25 | 0.25 |
| Studie Ist / Soll Analyse | Justin Winistörfer | 1.50 | 1.25 |
| Netzplan | Justin Winistörfer | 0.50 | 1 |
| Arbeitsjournal | Justin Winistörfer | 0.50 | 0.75 |
| Expertenbesuch und Protokoll | Justin Winistörfer, Alex Krämer, Mathias Hirth | 2 | 2 |
| Zeitplan anpassen | Justin Winistörfer | 0 | 0.50 |
| **Gesamtzeit** | **-** | **8** | **9.25** |

**Tagesablauf:**

Heute konnte ich gut starten, da ich noch im Zeitplan bin. Als erstes habe ich begonnen über die Projektmethode zu schreiben, welche ich für die IPA verwende. Die verwendete Methode ist hierbei HERMES 5.1. Ich habe danach die notwendigen Sachen geschrieben wie: Szenario, Module und das ganze HERMES Zeug. Es war zuerst etwas schwer in HERMES hineinzukommen, da ich nur mit Scrum oder Kanban arbeite. Jedoch konnte ich mit meiner Hilfestellung, mich anschliessend zurecht finden. Jedoch hat dies doch mehr Zeit gekostet, als ich zuerst dachte.  
  
Als dies erledigt war, habe ich danach die notwendigen Organigramme und weitere Grafiken erstellt, welche ich für das Kapitel Projektorganisation und Projektrollen benötige. Dieser Abschnitt war recht selbsterklärend und war dementsprechend auch schneller fertig als ich dachte. Praktisch ist auch hier wieder, das wir noch viele Vorlagen von vorherigen IPAs haben. Jedoch musste ich mich kurz in HERMES einlesen, wegen den Projektrollen.

Somit konnte ich danach meine Reserve für den Formalen Teil 1, für das nochmalige lesen und prüfen auf Rechtschreibefehler nutzen, statt für ungeplante andere Dinge. Zum Glück habe ich dies auch gemacht, ich hatte nämlich noch ein paar Rechtschreibefehler gefunden.  
  
Der Abschluss des Formalen Teils zeichnet sich dann dadurch aus, dass ich nochmals den ganzen Teil gelesen und mit den Kriterien verglichen habe. Ich denke mal das wird passen. Ansonsten sehen wir es nochmals am 7.5.

Nun konnte ich endlich in den Teil 2 einsteigen und habe begonnen die Einleitung zu schreiben. Ich bin recht froh, dass ich für heute wohl den HERMES Teil hinter mir haben. Die Einleitung lief problemlos und ich habe soviel Zeit wie geschätzt benötigt.

Da ich gerade noch übrige Zeit hatte und der Expertenbesuch erst um 15:00 Uhr war, habe ich noch den Netzplan begonnen und auch gleich abgeschlossen. Ursprünglich sollte dieser auch heute fertig werden, jedoch mussten Änderungen am Zeitplan vorgenommen werden, weswegen der Netzplan eigentlich erst Morgen fertig sein sollte.

Danach kam anschliessend Herr Krämer. Zuerst war ich sehr nervös, da ich nicht wusste was mich erwarten würde, doch es stellte sich herraus, dass Herr Krämer recht entspannt ist, jedoch sehr viel Wert auf die Kriterien legt. Er hat mir dann während dem Besuch wertvolle Tipps gegeben, jedoch hat er mich auch auf Sachen hingewiesen, welche noch angepasst werden müssen. Grundsätzlich empfand ich den Besuch als entspannt und hat mir das Gefühl bekräftigt, dass ich die IPA bestehen werde.  
  
Kommen wir noch zur nervigsten und ersten ungeplannten Tätigkeit, welche ich vollbringen musste. Beim Expertenbesuch stellte sich herraus, dass im Zeitplan noch unstimmigkeiten sind und ich diese unbedingt anpassen müsste. So sollte die SOLL Arbeitszeit auf 80h liege und jeden Tag sollten genau 8h eingeplant werden. Somit musste ich zuerst eine alte Kopie abspeichern, und den Zeitplan überarbeiten. Also habe ich zuerst geschaut, wo ich einkürzen kann und wie ich die 80h gleichmässig verteilen kann. Grundsätzlich war diese Arbeit schnell erledigt, jedoch kam mir die Verwendung von Excel wieder ins Verhängnis und es war mühsam alles anzupassen, jedoch wurde ich dann doch noch fertig.

**Hilfestellungen:**

Mit der HERMES Dokumentation von <https://hermes.admin.ch/> habe ich mir einen tieferen Einblick in die ganze Projektmethodik verschafft, damit ich dies bei meinen Kaptieln auch anwenden kann und allgemein bei meiner gesamten IPA.

**Reflexion:**

Das lief gut:

Grundsätzlich waren die ganze Tipparbeit schnell und sauber abgeschlossen und ich war froh, dass ich meine Reserve nutzen konnte zum selbst entgegenlesen. Auch den Expertenbesuch würde ich als erfolgreich und angenehm kennzeichnen. Ich denke die ganze Tipparbeit ist für mich zwar mühsam, jedoch erledige ich diese meist schnell, da ich behaupten würde, dass ich gewandt mit Worten umgehen kann und auch schnell tippen.

Das lief nicht so gut:  
Zuerst hatte ich Mühe mich mit Hermes einzuarbeiten, besonders da es kein passendes Szenario für meine IPA gab. Somit musste ich ein individuelles Szenario erstellen, welches jedoch auf dem Szenario «IT-Standardanwendung» basiert. So konnte ich danach gewisse Module und Meilensteine streichen, damit das individuelle Szenario auf meine Arbeit abgestimmt ist. Ich kann mir leider nicht erklären wieso ich HERMES gewählt habe. Jedoch dachte ich, dass ich dies einfach genommen habe, da es meist jeder so macht und man auch recht strikt an das Modell gebunden ist. Es war halt gerade die einfachste Möglichkeit, wobei bestimmt Kanban auch eine sehr interessante Methode gewesen wäre für die IPA.   
  
Der Zeitplan ist ein weiterer Punkt, welche mir kurz vor Feierabend nochmals dazwischen gefunkt hat. Es hat mir wieder gezeigt, dass Excel eventuell doch kein sinnvolles Tool ist, um einen Zeitplan zu erstellen. Es war etwas mühselig, jedoch schnell erledigt und ich denke, dass die Anpassungen schlussendlich wichtig waren, da es mich ansonsten Punkte eingebüsst hätte.   
  
Meine heutigen Erkenntnisse:  
Wenn ich es kurz fassen würde. Kein Benutzen von Excel für einen Zeitplan. HERMES ist recht Modular und kann sehr gut auf eigene Bedürfnisse angepasst werden. Auch wenn ich kein Fan von HERMES bin, denke ich, dass auch HERMES seine da-seins-Berechtigung hat.  
  
**Meine Schlussbilanz:**

Ich bin recht zufrieden mit dem heutigen Tag, auch wenn mir ein ungeplanter Task in die quere kam, welcher mich Zeit gekostet hat. Ich bin jedoch froh darum, dass ich morgen wieder nach Zeitplan arbeiten kann.

**Nächste Schritte:**Da ich für den heutigen Tag, im Zeitplan noch dabei bin, kann ich morgen nach Zeitplan weiterarbeiten. Wichtig ist jedoch, dass ich mein alter und neuer Zeitplan morgen bis um 12:00 Uhr auf PkOrg in den Dokumentenpool hochlade.

## Arbeitsjournal vom 24.04.2024

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte Personen | SOLL Zeit (h) | IST Zeit (h) |
| Netzplan | Justin Winistörfer | 0.50 | 0 |
| Projekt- und Vorgehensziele | Justin Winistörfer | 1.50 | 1.25 |
| Anforderungen | Justin Winistörfer | 1 | 1 |
| Reserve Initialisierung | Justin Winistörfer | 0.50 | 0.50 |
| Projektfreigabe | Justin Winistörfer, Mathias Hirth | 0.25 | 0.25 |
| Systemkonzept | Justin Winistörfer | 1 | 1 |
| Namenskonzept | Justin Winistörfer | 0.50 | 0.50 |
| Netzwerkkonzept | Justin Winistörfer | 0.75 | 0.75 |
| Berechtigungskonzept | Justin Winistörfer | 0.50 | 0.50 |
| ISDS-Konzept | Justin Winistörfer | 0.75 | 0.75 |
| Testkonzept | Justin Winistörfer | 0.25 | 0.25 |
| Lernjournal | Justin Winistörfer | 0.5 | 0.5 |
| **Gesamtzeit** | **-** | **8** | **7.25** |

**Tagesablauf:**

Heute konnte ich mir am Morgen bereits schon eine halbe Stunde sparen, da ich den geplanten Netzplan bereits gestern schon fertig gestellt habe. Der Morgen beginnte somit recht gut. Also habe ich heute als erstes die Projekt- und Vorgehensziele geschrieben, in welchem ich mir zuerst mal ein paar Gedanken machen musste, was ich mir als Ziel überhaupt setzen möchte. Die Ziele habe ich danach anschliessend SMART definiert. Dieser Task war tatsächlich auch schneller erledigt als gedacht, da es aufgrund von den vielen Kriterien eigentlich recht einfach ist, die Ziele zu definieren. Man erstellte sie einfach aufgrund der Kriterien. Anschliessend habe ich direkt die Anforderungen für das Projekt anhand der Ziele geschrieben, was eigentlich auch nichts grossartiges war. Ich hatte das Glück, dass ich solche Zieldefinitionen schon oft genug angeschaut habe und auch daraus schliessende Anforderungen.  
  
Dadurch, dass ich keine Zeitrückstände mit den Tasks hatte, konnte ich die geplante Reservezeit wieder voll und ganz nutzen, um Rechtschreibe- und Grammatikfehler auszubessern, was definitiv auch nötig war. Das freut mich immer am meisten, wenn ich die Reserve für so etwas nutzen kann und nicht um meinen Zeitplan aufzuholen. Nachdem alles ausgebessert war, habe ich Mathias das Dokument überreicht, um mir die Projektfreigabe zu geben. Da war alles ok.

Nun kam ich zum nächsten langwierigen Theorieteil: Die Konzeptphase. Hier habe ich heute besonders viele Konzepte geschrieben. Das Systemkonzept und Namenskonzept waren recht schnell gemacht, da ich AWS in- und auswendig kenne und auch ein Namenskonzept recht selbsterklärend ist. Das Netzwerkkonzept hatte mich zwar wieder etwas mehr Zeit gekostet, jedoch nur, weil ich ständig neue Tabellen einsetzen musste, was etwas mühsam war, dieses viele Klicken.

Nach dem Mittag habe ich dann versucht das Berechtigungskonzept zu erstellen, jedoch hat sich dieser Teil für mich nach einer Ewigkeit angefühlt, da ich überhaupt keine Energie hatte. Jedoch konnte ich auch hier schnell ein paar Tabelleneinfügen. Viel mühsamer wurde dann das ISDS-Konzept. Dies weil es mir schwer gefallen ist mir Gefahren für ein Projekt zu überlegen, welches ich noch überhaupt nicht umgesetzt habe. Eigentlich hätte ich hier gar nicht die ganze Zeit gebraucht, jedoch waren eben genau diese Gefahren für mich ein kleiner Knackpunkt. Wichtig war mir einfach, dass ich im Zeitplan bleibe. Nach dem ISDS-Konzept hatte ich noch kurz die Einleitung für das Testkonzept geschrieben, für mehr hat es dann heute schlussendlich nicht gereicht.

**Hilfestellungen:**

Mit der HERMES Dokumentation von <https://hermes.admin.ch/> habe ich mich besonders in die Punkte: Ziele, Systemkonzept und ISDS-Konzept eingelesen, damit ich dies nach der HERMES-Norm machen kann. Für die anderen Konzepte gab es keine Vorgaben.

**Reflexion:**

Das lief gut:

Ich bin sehr froh darüber, dass ich aktuell immer noch im Zeitplan mitmache, obwohl ich bereits schon etwas mehr IST-Zeit habe, jedoch hänge ich nicht hinterher und das ist mir besonders wichtig. Ich denke grundsätzlich die ganzen Konzepte, bis auf ISDS und Berechtigungskonzept waren rasch erledigt, da ich solche Konzepte nicht zum erstenmal schreibe. Dies spart mir so natürlich viel Zeit ein. Besonders wenn es um AWS geht, bekomme ich immer einen Motivationsschub, da ich ein ganzes Buch darüber schreiben könnte. Ich musste mich recht zusammenreisen, das ich da nicht übertreibe.

Das lief nicht so gut:  
Wie bereits erwähnt, hat mir die fehlende Energie heute etwas zu schaffen gemacht und es war anstrengend am Ball zu bleiben. Leider kann ich mir nicht erklären, wieso dies so geschehen ist. Teilweise war es am Nachmittag nicht einfach motiviert und konzentriert zu bleiben. Ich habe mich immer wieder erwischt, dass ich eine Pause machen wollte oder mich einfach ablenken wollte. Jedoch weiss ich, dass der Zeitplan nicht auf mich wartet, deswegen konnte ich diese Situation schnell abwenden.  
  
Meine heutigen Erkenntnisse:  
HERMES hat so viele theoretische Teile und es macht die ganze Projektarbeit, besonders zum Beginn, etwas träge. Ich weiss nicht ob ich überhaupt wieder HERMES verwenden würde, wenn es nicht vorgegeben wäre. Jedoch ist es für mich auch heute wieder ein guter Input, da ich heute wieder ein paar Sachen über HERMES gelernt habe. Und ich denke auch, dass die Konzepte die IPA um vieles erleichtern werden, da du dann nur noch nach Vorgabe umsetzen musst.  
  
**Meine Schlussbilanz:**

Ich bin froh, dass ich den heutigen Tag hintermir habe und ich Morgen eine Pause aufgrund der Schule einlegen kann. Jedoch denke ich, dass ich heute einen erfolgreicheren Tag hatte als gestern und ich immernoch überzeugt bin, dass ich die IPA definitiv bestehen werden.

**Nächste Schritte:**Als nächstes muss ich nur noch die zwei letzten Konzepte schreiben und danach sollte ich am Freitag endlich mit der Realisierung beginnen können. Ich bin schon sehr gespannt, wie das laufen wird. Ich hoffe, dass keine zu grossen Probleme auftreten werden.

## Arbeitsjournal vom 26.04.2024

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Beteiligte Personen | SOLL Zeit (h) | IST Zeit (h) |
| Testkonzept | Justin Winistörfer | 1.00 | 1.50 |
| Automatisierungskonzept |  | 0.75 | 0.50 |
| Phasenfreigabe |  | 0.25 | 0.25 |
| IDE Umgebung bereitstellen |  | 1 | 0.75 |
| Zugang zu AWS bereitstellen |  | 1.25 | 1 |
| Terraform Scripts zur Bereitstellung der grundlegenden AWS Services erstellen |  | 2.75 | 2.75 |
| Terraform Scripts zur Bereitstellung des EKS erstellen |  | 0.50 | 0.50 |
| Lernjournal |  | 0.50 | 0.50 |
| **Gesamtzeit** | **-** | **8** | **7.75** |

**Tagesablauf:**

Heute war ich endlich auf dem Endspurt von der Konzeptphase, auch wenn ich zuerst etwas Mühe hatte, da ich schon mit Kopfschmerzen am Morgen aufgewacht bin. Um die Konzeptphase bald hinter mich zu bringen, habe ich zuerst das Testkonzept fertig geschrieben. Dies hat zwar etwas länger gedauert, da ich mir besonders viel Zeit genommen habe, die Abgrenzungen zu schreiben, da mir dies am Expertenbesuchstag sehr geraten wurde. Dementsprechend hatte ich dort länger als geplant. Als das Konzept fertig war, habe ich noch das Automatisierungskonzept geschrieben und dies war gar nicht so einfach, da für mich die Pipelines ziemlich neu sind, da ich jedoch trotzdem weiss wie man diese in der Theorie umgesetzt werden, konnte ich dies anhand meines theoretischen Wissens machen und war dabei auch noch schneller fertig als geplant. Als diese Konzepte fertig waren, habe ich mit Mathias die Phasenfreigabe gemacht, damit ich endlich mit der Realisierung starten kann, dieser Moment, auf den ich die letzten 4 Tage gewartet habe.  
  
Die Realisierung war heute noch sehr angenehm, da ich heute nur paar Vorbereitungen machen musste und dann mal mit den ersten Scripts, welche für mich selbsterklärend sind, erstellen musste. Somit war besonders die IDE Umgebung bereitstellen, schnell erledigt und dokumentiert. Der Zugang zu AWS war auch schneller erledigt als geplant, da ich noch ein altes Script hatte, welches ich nur auf meine aktuellen Credentials anpassen musste. Dies hatte auch sehr einfach funktioniert. Die Terraform-Scripts selbst zum bereitstellen der Infrastruktur waren auch sehr einfach erstellt, da es von Terraform eine sehr gute Dokumentation gibt und ich auch schon mit Terraform gearbeitet habe, dies war nichts neues für mich. Ich könnte mir jedoch vorstellen, dass es dann mit dem EKS etwas anstrengender werden könnte. Mit diesen Scripts für das EKS habe ich zwar heute auch schon begonnen, jedoch war dies auch noch einfach und noch nicht Kubernetes spezifisch.

**Hilfestellungen:**

Mit der HERMES Dokumentation von <https://hermes.admin.ch/> habe ich mich in den Punkt «Testkonzept» eingelesen und einen Blick in die Vorlage geworfen, damit ich weiss, wie ich mein Testkonzept aufbauen muss. Ebenso habe ich später in der Realisierung die Dokumentation von Terraform zur Hand genommen, damit ich weiss, wie ich meine Terraform-Scripts bauen muss, diese Dokumentation ist hier ersichtlich: <https://registry.terraform.io/providers/hashicorp/aws/latest/docs>

**Reflexion:**

Das lief gut:

ENDLICH habe ich die Konzeptphase hinter mir und ich konnte endlich mit der Realisierung anfangen. Diese lief heute auch noch ziemlich gut und ich bin sehr zufrieden, das heute ansonsten alles nach Plan lief und ich keine grossen Probleme hatte.  
  
Das lief nicht so gut:  
Es gab heute nichts wirklich, dass nicht so gut lief. Das einzige Hindernis war lediglich, dass ich etwas länger mit dem Testkonzept gebraucht hatte, aber ansonsten sehe ich für heute keine Mängel.  
  
Meine heutigen Erkenntnisse:  
Es ist erstaunlich zu sehen, wie viel Zeit man für Aufgaben braucht, welche man normalerweise öfters ausführt. Man achtet sich normalerweise gar nicht wirklich auf die Zeit, dies ist jedoch bei der IPA ganz anders und achtet sich extrem und plötzlich sieht man selbst, dass man «viel» Zeit für eine Aufgabe benötigt, obwohl das Gefühl etwas anderes sagt. Deswegen bin ich auch sehr froh, dass ich mir so viel Zeit beim Zeitplan genommen habe, da ich so, bis jetzt, meist sehr gut mit der Zeit lag, bis auf ein paar Ausnahmen.  
  
**Meine Schlussbilanz:**

Es ist gut zu wissen, dass man auch nach Tag 4 immer noch im Zeitplan ist und dass ich endlich mit der Realisierung begonnen hatte. Ich denke wichtig mitzunehmen ist, dass man meist die Zeit unterschätzt bei alltäglichen Aufgaben und man mehr Zeit benötigt, als einem bewusst ist.

**Nächste Schritte:**Am Montag werde ich mit den Scripts für meine Infrastruktur, spezifisch für EKS, weiterfahren und mal sehen, wie das läuft.

# Teil 2: Individueller Praktischer Teil

# Einleitung

## T-Systems Schweiz AG

Die T-Systems International ist ein internationales Tochterunternehmen des Unternehmens Deutschen Telekom. Einen Standort hat die T-Systems International ebenfalls in der Schweiz, unter dem Namen T-Systems Schweiz AG. Die T-Systems Schweiz AG hat sich dabei besonders auf verschiedene Informatikdienstleistungen, für Grosskunden, in diversen Branchengebieten spezialisiert.

COMMENT: Bild vom Standort machen und einfügen

Jährlich werden von der T-Systems, in der Schweiz, mehrere Lehrlinge und Praktikanten, im Bereich der Informatik, sowie auch im kaufmännischen Bereich, ausgebildet. Das Besondere dabei an der Informatiklehre ist, dass die Lehrlinge im Basislehrjahr nicht in einem externen Betrieb ausgebildet werden, sondern intern in dem sogenannten «Berufsbildungsinstitut». In diesem Basislehrjahr lernen die Lehrlinge bereits schon früh den Umgang mit Kunden und die Verantwortung in der internen IT kennen.

Dies umfasst besonders das selbstständige Erarbeiten von Aufträgen und die Kundenkommunikation mit dem internen Mitarbeiter, welche auf die interne IT angewiesen sind.

Sobald dieses Basislehrjahr abgeschlossen wurde, wird der Lehrling in eine fachspezifische Abteilung eingeteilt, in welchem er die nötigen Fachkenntnisse erlernt und anwenden kann.

## Core SERVICES

Mittlerweile hat sich die T-Systems aber auf FlexOrg umstrukturiert. Dies bedeutet, dass es nur noch aus organisatorischen Gründen diverse Abteilungen gibt, jedoch innerhalb dieser Abteilungen keine festen Teams mehr gibt.

Aktuell befinde ich mich in der organisatorischen Abteilung «Core SERVICES», in welchem ich für das «Team TAU» arbeite. Im Team TAU arbeiten wir für einen Grosskunden, an einer Multi-Cloud Applikation, für welches ich hauptsächlich an einem AngularJS Frontend arbeite.

Diese Multi-Cloud-Applikation ermöglicht es, zentral VMs auf verschiedenen Cloudplattformen zu erstellen und diese anschliessend ebenso zentral zu verwalten.

## Begründung für die IPA-Themen Wahl

Aktuell sind wir dabei unsere eigene Cloudinfrastruktur aufzubauen, welche den Namen «Swiss Open Telekom Cloud», kurz «SOTC», trägt. Durch den Aufbau unserer Cloud und ebenso gleichzeitig die Migration von alter Infrastruktur auf unsere eigene Cloud, gab es in letzter Zeit viele Systemausfälle, welche zu erheblichen Unterbrechungen führten. Schlicht und einfach, befindet sich unsere Cloud noch nicht in einem stabilen Zustand, und es werden stets wieder Änderungen vorgenommen.

Damit wir solche Ausfälle vorbeugen können und unseren Systemen mehr Resilienz bieten können, möchten wir das Tool «Chaos-Mesh» in Betrieb nehmen, damit wir besonders unsere Microservices auf Worst-Case Szenarien härten können.

Bevor wir jedoch diese Applikation in Betrieb nehmen können, wurde für diese IPA genau dieses Tool vorgegeben, damit dieses während der IPA auf ihre Funktionalität und Integrität getestet werden kann. So soll nicht nur einfaches Chaos-Engineering möglich sein, sondern das Anwenden und Testen solcher Chaos-Experimente, soll auch gleich automatisiert erfolgen.

Nach Abschluss dieser IPA sollen Erkenntnisse an das eigene Team weitergegeben werden, damit diese anschliessend anhand meiner Abschlussarbeit sich entscheiden können, ob man das Tool in den Betrieb aufnehmen möchte.

So können wir für künftige Anwendungen, welche wir auf unserer eigenen Cloud bereitstellen, diverse automatische Tests vornehmen, um zu prüfen, ob das System resilient genug ist oder ob weitere Anpassungen vorgenommen werden müssen.

# Initialisierung

Mit der Initalisierungsphase beginnt auch der Start des Projektes, welche eine Ausgangslage für das Projekt definiert, und sicherstellt, dass die erstellten Projektziele und Anforderungen eingehalten werden. Am Ende dieser Phase wird geprüft, ob eine Freigebung des Projektes als sinnvoll erscheint.



Abbildung 10: Initialisierungsphase

## Ist- / Soll-Analyse

Die nachfolgende Ist- / Soll-Analyse, dient für dieses Projekt als aktuellen Anhaltspunkt, damit die persönlichen, sowie Projektziele definiert und später auch umgesetzt werden können.

### Ist-Zustand

Aktuelle werden alle unsere Applikationen, welche wir den Kunden zur Verfügung stellen, auf einer beliebigen Cloudplattform in einem Kubernetes-Cluster bereitgestellt. Dabei werden keine sogenannten „Chaos-Experimente“ durchgeführt, zum Prüfen wie widerstandsfähig die bereitgestellte Applikation tatsächlich ist. Es werden lediglich durch automatisierte Tests, die Funktionalitäten der jeweiligen Applikationen geprüft.  
  
Im folgenden Netzplan wird eine Umgebung in der AWS Cloud nachgestellt, welche aufzeigt, wie eine Applikation von uns beispielsweise bereitgestellt wird:  
  
Ein Bild, das Text, Screenshot, Multimedia-Software, Software enthält.

Automatisch generierte Beschreibung

Abbildung 11: Übersicht Ist-Zustand in der AWS Umgebung

Auf dieser Abbildung ist ein Diagramm ersichtlich, welches die wichtigsten AWS Services aufzeigt und wie diese zusammen in Verbindung stehen. So sieht man, dass Anfragen, welche von einem Client aus dem Internet kommen, über den Application Load Balancer, an den AWS EKS gesendet werden. In den späteren Kapiteln werden die genaue Funktionsweise und weitere Erklärungen zu diesen Services geschildert.

Da diese Grafik jetzt noch nicht aussagekräftig genug ist, um den Stand über die bereitgestellte Applikation zu erfahren, folgt nun eine weitere Abbildung, welche einen Einblick in AWS EKS ermöglicht.

Ein Bild, das Text, Screenshot, Rechteck, Design enthält.

Automatisch generierte Beschreibung

Abbildung 12: Übersicht Ist-Zustand innerhalb eines AWS EKS

In dieser Abbildung erkennt man, wie das Innenleben in AWS EKS in etwa aussehen könnte. Dabei sieht man, dass in diesem Cluster zwei verschiedene Namespaces aufgesetzt sind, damit die Monitoring Applikationen, logisch, von der effektiven Applikation getrennt ist, welche wir für den Kunden zur Verfügung stellen.

Im Namespace namens «application-ns» werden dabei alle nötigen Komponenten bereitgestellt, welche für den Betrieb der Applikation notwendig sind. Im Namespace «monitoring-ns» werden meist zwei verschiedene Tools eingesetzt, damit die bereitgestellte Applikation überwacht werden kann.

Diese beiden Abbildungen sind jeweils sehr vereinfachte Beispiele, denn in der Regel befinden sich noch viele weitere Services und Namespaces in einer AWS Umgebung. Jedoch sieht die Integration von Chaos-Mesh in allen Szenarien gleich aus, weswegen auf die weitere Komponente verzichtet werden kann und man sich auf die vitalen Komponenten konzentrieren kann.

### Soll-Zustand

Damit obengenanntes Problem in Zukunft vermieden werden kann, müsste das Tool „Chaos-Mesh“ in die jeweiligen Kubernetes-Umgebungen integriert werden.

Dies könnte anhand des vorherigen Beispiels innerhalb des EKS wie folgt aussehen:

Ein Bild, das Text, Screenshot, Display, Computersymbol enthält.

Automatisch generierte Beschreibung

Abbildung 13: Soll-Zustand innerhalb des EKS

Man beachte nun dabei, dass nur ein weiterer Namespace zum EKS hinzugefügt wurde, welche den Namen „chaos-mesh-ns“ besitzt. In diesem Namespace werden danach alle nötigen Pods gehostet, welche für den Betrieb von Chaos-Mesh nötig sind. Ebenso besteht die Möglichkeit zur Kommunikation zwischen den beiden Namespaces, welche später essenziel wird.

Dadurch das nur ein weiterer Namespace in die bestehende Umgebung integriert werden muss, müssen ausserhalb oder sogar innerhalb des EKS keine weiteren Änderungen vorgenommen werden, was die Integration von Chaos-Mesh, als sehr leicht darstellt.

In diesem Projekt soll Chaos-Mesh in einer fiktiven Cloud-Native Kubernetes Umgebung eingesetzt werden, welche ebenso auf der AWS Cloud bereitgestellt wird, damit eine Integration in eine bestehende Umgebung simuliert werden kann.

Damit eine möglichst identische Integration nachgeahmt werden kann, wird für dieses Projekt, das obige Diagramm übernommen, mit der Ausnahme, dass sich kein Database Deployment im Namespace „application-ns“ befindet und der Namespace „monitoring-ns“ auch komplett aus der Umgebung entfernt wird. Das Database Deployment wird dabei auf den Service „MongoDB Atlas“ ausgelagert. Diese Massnahmen wurden deswegen getroffen, damit Ressourcen und Zeit eingespart werden können, und das Projekt auch innerhalb der vorgegebenen 80 Stunden umgesetzt werden kann. Zudem kann mit der Auslagerung der Datenbank, weitere externe Abhängigkeiten getestet und deren Reaktion ausgewertet werden.

## Ziele

Da die Initialisierungsphase ebenso das Setzen von Zielen vorsieht, werden in diesem Kapitel Projektziele, sowie auch persönliche Vorgehensziele festgelegt.

### SMART

Alle Ziele innerhalb dieses Projektes werden nach SMART definiert. Das Akronym für SMART bildet sich dabei aus folgenden Bestandteilen, bei der Zielsetzung:[[7]](#footnote-7)  
  
S – Ziele müssen spezifisch sein

M – Ziele müssen messbar sein

A – Ziele müssen erreichbar (engl. achievable) sein

R – Ziele müssen realisierbar sein

T – Ziele müssen terminierbar sein

### Persönliche Vorgehensziele

In der nachfolgenden Tabelle werden meine persönlichen Vorgehensziele gesetzt und erläuter.

|  |  |  |
| --- | --- | --- |
| Ziel Nr. | Ziel | Termin |
| PVZ-1 | Alle Meilensteine werden termingerecht gemäss Zeitplan eingehalten | 07.05.2024 |
| PVZ-2 | Der Abgabetermin der IPA wird eingehalten | 07.05.2024 |
| PVZ-3 | Die Umsetzung von HERMES 5.1 folgt korrekt gemäss Vorgaben | 07.05.2024 |
| PVZ-4 | Das Lernjournal wird jeden Tag geführt | 07.05.2024 |
| PVZ-5 | Die Dokumentation wird jeden Abend auf beiden Speichermedien gesichert und das Versionierungsschema wird eingehalten. | 07.05.2024 |

### Projektziele

In der untenstehenden Tabelle werden die Projektziele für die IPA definiert.

|  |  |  |
| --- | --- | --- |
| Ziel Nr. | Ziel | Termin |
| Z-1 | Die vollständige Infrastruktur wird, gemäss Zeitplan, termingerecht bereitgestellt, um Verzögerung des Projekts zu vermeiden. | 07.05.2024 |
| Z-2 | Das Benutzerhandbuch stellt sicher, dass die Anleitung den User in jeder Situation zum Erfolg bringt. | 07.05.2024 |
| Z-3 | Es werden alle Experimente in der IPA per YAML umgesetzt und angewendet, damit eine anschliessende Auswertung möglich ist. | 07.05.2024 |
| Z-4 | Das CloudWatch Dashboard ist funktionstüchtig und zeigt alle nötigen und wichtigen Metriken an. | 07.05.2024 |
| Z-5 | Mindestens ein Experiment wird über die GitHub Actions Pipeline automatisiert. | 07.05.2024 |
| Z-6 | Die Realisierungsphase muss, gemäss Zeitplan, termingerecht abgeschlossen werden. | 07.05.2024 |
| Z-7 | Chaos-Mesh wird gemäss offizieller Dokumentation per Helm installiert. | 07.05.2024 |
| Z-8 | Der Terraform State wird stets in einem AWS S3 Bucket gespeichert. | 07.05.2024 |
| Z-9 | Es soll jederzeit ein Tagesaktuellerstand, aller verwendeten und erstellten Scripts, auf GitHub gesichert sein. | 07.05.2024 |

## Anforderungen

Anhand der zuvor definierten Projektziele, werden in der nachfolgenden Tabelle die Anforderungen für das Projekt definiert und beschrieben.

|  |  |  |
| --- | --- | --- |
| Anforderung Nr. | Ziel Nr. | Beschreibung |
| A-1 | Z-1 | Infrastruktur gemäss Zeitplan bereitstellen, umsetzen und testen. |
| A-2 | Z-1 | Regelmässige Dokumentationseinträge über die Infrastruktur führen. |
| A-3 | Z-2 | Erstellung und Prüfung des Benutzerhandbuchs. |
| A-4 | Z-2 | Validierung des Handbuchs durch Benutzertests. |
| A-5 | Z-3 | Durchführung und Überprüfung aller YAML-basierten Experimente. |
| A-6 | Z-4 | Entwicklung und Konfiguration des CloudWatch Dashboards zur Überwachung der erforderlichen Metriken. |
| A-7 | Z-4 | Funktionstests und Validierung des Dashboards. |
| A-8 | Z-5 | Implementierung der GitHub Actions Pipeline für mindestens ein Experiment. |
| A-9 | Z-5 | Test und Überprüfung der Automatisierung des Experiments. |
| A-10 | Z-7 | Installation von Chaos-Mesh mittels Helm gemäss der offiziellen Anleitung. |
| A-11 | Z-8 | Sicherstellung der kontinuierlichen Aktualisierung und Sicherheit des Terraform State. |

## Variantenentscheid

HERMES 5.1 sieht normalerweise vor, dass für ein Projekt verschiedene Varianten gewählt und verglichen werden, damit anschliessend ein Variantenentscheid anhand der Ziele und Anforderungen getroffen werde kann. Da jedoch in diesem Projekt das Tool für den Einsatz im Projekt vorgegeben ist, kann hier kein Variantenentscheid getroffen werden.  
  
Damit jedoch trotzdem ein Überblick für künftige Chaos-Engineering Projekte verschafft werden kann, werden nachfolgend ein paar Alternativen zu Chaos-Mesh aufgelistet:

* Chaos Monkey (<https://netflix.github.io/chaosmonkey/>)
* Litmus (<https://litmuschaos.io/>)
* SteadyBit (<https://steadybit.com/>)

## Projektfreigabe

Durch die Unterschrift des Auftraggebers und des Projektleiters wird festgelegt, dass die Initialisierungsphase abgeschlossen wurde und somit die Konzeptphase freigegeben ist.

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Name | Rolle | Unterschrift |
| 24.04.2024 | Justin Winistörfer | Projektleiter |  |
| 24.04.2024 | Mathias Hirth | Auftraggeber |  |

# Konzept

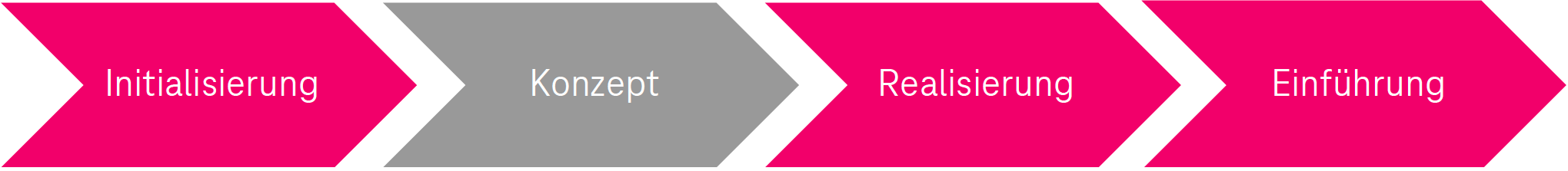
Nach der Projektfreigabe startet, laut HERMES 5.1, die Konzeptphase des Projektes. In diesem Kapitel werden nun verschiedene Konzepte erstellt, welche für die Realisierung danach notwendig sind. Sobald diese Phase durchlaufen ist, und eine Phasenfreigabe stattgefunden hat, wird mit der Realisierung und somit mit der Umsetzung des Projektes, anhand der Konzepte gestartet.  
  


Abbildung 14: Konzeptphase

## Systemkonzept

Das Systemkonzept dient als Vertiefung der Ist- / Soll-Analyse. So sollen Systemanforderungen konkretisiert werden und somit eine Systemarchitektur definiert und erstellt werden. Ebenso sollen dabei in einem Systemkonzept diverse Anforderungen erarbeitet und bewertet werden. Da jedoch anhand der Vorgaben und das Benutzen einer Cloudplattform kein Spielraum für diverse Varianten offensteht, werden in diesem Systemkonzept nicht verschiedene Varianten erarbeitet.

### AWS

In diesem Projekt wird zum Bereitstellen der Systemumgebung, die Cloud Plattform Amazon Web Services, kurz AWS, verwendet. AWS ist ein von Amazon gegründeter Cloud-Computing-Anbieter, welcher sich durch seine verschiedenen angebotenen Services hervorhebt. AWS zählt zu den grössten Anbietern, da sich diese Plattform für alle anbietet, sei es natürliche bis hin zu grossen juristischen Personen. Auch wir, in der T-Systems, verwenden AWS zu einem sehr grossen Teil, zum Bereitstellen von Kundenapplikation oder auch virtuelle Infrastrukturen.

Der Service, welcher als Kernpunkt dieses Projektes agiert, ist der AWS Elastic Kubernetes Service (AWS EKS). Wie es der Name bereits schon verrät, kann mit diesem Service eine komplette Kubernetes Umgebung über die Cloud bereitgestellt werden. Dadurch das dieser Service vollständig auf einer Cloud basiert, wird eine solche Kubernetesumgebung „Cloud-Native Kubernetes Umgebung“ genannt. Cloud-Native bezeichnet im Grunde alles, was vollständig in der Cloud bereitgestellt wird und auch für die Cloud erstellt wurde.

Da jedoch AWS EKS nicht ausreicht, um eine externe Kommunikation mit dem Internet zu ermöglichen, müssen noch folgende Services angewendet werden:

* VPC (Virtual Private Cloud)
* IGW (Internet-Gateway)
* Public & Private Subnets
* ALB (Application Load Balancer)
* NAT Gateways
* RT (Route Tables)
* EKS (Elastic Kubernetes Service)
* IAM
* S3 (Simple Storage Service)

Setzt man all diese Services gemeinsam ein, so würde die Architektur danach wie folgt aussehen:

Ein Bild, das Text, Screenshot, Software, Multimedia enthält.

Automatisch generierte Beschreibung

Abbildung 15: AWS Systemarchitektur

Auf dieser Abbildung sind nun alle erwähnten Services, mit Ausnahme der Route Tables und dem S3 Bucket abgebildet. Die beiden erwähnten Services werden dabei nicht abgebildet, da die Route Tables jeweils mit einem Subnet assoziiert werden und der S3 Bucket nicht im eigenen VPC gehostet wird. Jeder Service erfüllt dabei seine eigene Aufgabe, in der nachfolgenden Tabelle, werden die jeweiligen Aufgaben kurz erklärt:

|  |  |
| --- | --- |
| Service | Beschreibung |
| AWS VPC | Die Virtual Private Cloud stellt über die Cloud einen eigenen virtuellen Bereich dar. Dies ist sehr vergleichbar mit einem virtuellen Rechenzentrum, in welchem man anschliessend alles deployen kann. |
| AWS IGW | Der Internet-Gateway ist zuständig, dass eine Verbindung mit der Aussenwelt (Internet) hergestellt werden kann. Ohne IGW ist auch keine Kommunikation nach ausserhalb der AWS Plattform möglich. |
| Public & Private Subnets | Dies sind normale Subnetze, wie man sie auch aus dem Netzwerk-Bereich kennt, jedoch virtuell. |
| AWS ALB | Der Application Load Balancer dient als dynamischer Load Balancer, welcher den einkommenden Traffic gleichmässig auf die Pods innerhalb von AWS EKS verteilt. |
| NAT Gateways | Auch hier sind die NAT Gateways, normale NAT Gateways, welche eine „private“ Verbindung ins Internet ermöglichen. Dies ist besonders möglich, wenn ein Service ins Internet zugreifen muss, jedoch nicht öffentlich erreichbar sein soll. |
| RT | Die AWS Route Tables sind auch hier, wie im physischen Netzwerk üblich, Routing Tabellen, welche verschiedene Regeln enthalten. Der Unterschied dabei ist, dass die Route Tables jeweils direkt mit einem Subnet assoziiert werden. |
| AWS EKS | AWS Elastic Kubernetes Services ist ein Service, welcher eine automatische Bereitstellung einer Kubernetes Umgebung erledigt. |
| AWS IAM | AWS Identitiy and Access Management ist ein Service, welcher die Erstellung von Nutzern, Policies und Gruppen ermöglicht. Vergleichbar mit den typischen Active Directories. |
| AWS S3 | AWS Simple Storage Service ist ein Objektspeicher, welcher die Dateiverwaltung in der Cloud ermöglicht. So können alle beliebigen Dateien in einem sogenannten Bucket gespeichert werden. |

Durch das Zusammenspiel all dieser Services, kann schnell, flexibel und effektiv eine Cloud-Native Systemarchitektur erstellt werden, auf welcher wir unsere Applikation bereitstellen können. Das praktische dabei ist, dass AWS dabei bereits schon viele Aufgaben abnimmt und man sehr intuitiv durch den Erstellungsprozess geleitet wird.

### Teilsysteme und Abhängigkeiten

Die ganze Projektumgebung besteht im Grunde, aus drei diversen und wichtigen Teilen: Die Infrastruktur auf AWS selbst, die Kubernetesumgebung mit der Applikation und Chaos-Mesh.

Dabei ist die Kubernetesumgebung von der darunterliegenden Infrastruktur abhängig, während Chaos-Mesh von der Kubernetesumgebung abhängig ist. Daraus lässt sich schliessen, dass die Infrastruktur einen besonders hohen Stellenwert hat, da von dieser das ganze Projekt abhängig ist.

## Namenskonzept

Damit die Namensgebung von Ressourcen identisch ist und nicht zur Verwirrung führen kann, wird im folgenden Abschnitt ein globales Namenskonzept erstellt, welches innerhalb dieses Projektes angewendet wird.

### AWS Ressourcen

Für alle Ressourcen und Services, welche innerhalb von AWS bereitgestellt werden, zählt folgende Namenskonvention:

|  |  |  |
| --- | --- | --- |
| Ressourcentyp | Namenskonvention | Beispiel |
| EKS Cluster | eks-<projekt>-<umgebung> | eks-chaos-ipa-test |
| EKS Node Group | eks-<projekt>-ng-<grösse>-<umgebung> | eks-chaos-ipa-ng-t4gsmall-test |
| S3 Bucket | <projekt>-<umgebung>-<nutzen> | chaos-ipa-test-terraformstate |
| VPC | vpc-<projekt>-<umgebung> | vpc-chaos-ipa-test |
| Subnet | <subnet>-<projekt>-<umgebung>-<az> | subnet-chaos-ipa-test-eu-west-1a |
| Security Group | sg-<projekt>-<umgebung>-<ressource> | sg-chaos-ipa-test-eks |
| IAM Rolle | role-<projekt>-<umgebung>-<zweck> | role-chaos-ipa-test-eks-zugriff |
| IAM Richtlinie | policy-<projekt>-<umgebung>-<ressource> | policy-chaos-ipa-test-s3 |
| Application Load Balancer | alb-<projekt>-<umgebung>-<anwendung> | alb-chaos-ipa-test-app |
| CloudWatch Dashboard | cw-<projekt>-<umgebung>-<ressource> | cw-chaos-ipa-test-eks |
| IGW | igw-<projekt>-<umgebung> | igw-chaos-ipa-test |
| NAT | nat-<projekt>-<umgebung> | nat-chaos-ipa-test |
| Elastic IP | eip-<projekt>-<umgebung> | eip-chaos-ipa-test |
| Route Table | route-<projekt>-<umgebung>-<subnet> | Route-chaos-ipa-test-private |

Erläuterungen zur Namenskonvention:

* <projekt> ist der Projektname oder ein spezifisches Kürzel, das das Projekt beschreibt, in diesem Projekt dementsprechend «chaos-ipa».
* <umgebung> bezieht sich auf die Umgebung wie prod (Produktion), dev (Entwicklung), test, usw.
* <zweck> oder <ressource> beschreibt den spezifischen Zweck oder die spezifische Ressource, z.B. zugriff (Zugriff), app (Anwendung), usw.
* <groesse> wird verwendet, um die Grösse oder Kapazität von Node Groups zu beschreiben, z.B. t4gsmall, t3micro, usw. Dabei handelt es sich um die Grössen, welche AWS anbietet.
* <az> steht für die AWS Availability Zone, z.B. eu-west-1a.
* <anwendung> bezieht sich auf die spezifische Anwendung, die durch die Ressource bereitgestellt wird, z.B. app (Anwendung), db (Datenbank), usw.
* <subnet> definiert, ob es um ein Öffentliches (public) oder um ein Privates (private) Subnetz handelt.

### Kubernetes Ressourcen

Für alle Ressourcen und Services, welche innerhalb von Kubernetes bereitgestellt werden, zählt folgende Namenskonvention:

|  |  |  |
| --- | --- | --- |
| Ressourcentyp | Namenskonvention | Beispiel |
| Namespace | ns-<projekt>-<umgebung>-<zweck> | ns-chaos-ipa-test-app |
| Deployment | deploy-<projekt>-<umgebung>-<anwendung> | deploy-test-frontend |
| Service | svc-<projekt>-<umgebung>-<anwendung> | svc-chaos-ipa-test-frontend |
| Pod | pod-<projekt>-<umgebung>-<anwendung>-<unique-id> | pod-chaos-ipa-test-frontend-1234 |
| Secret | secret-<projekt>-<umgebung>-<zweck> | secret-chaos-ipa-test-docker-credentials |
| Ingress | ingress-<projekt>-<umgebung>-<anwendung> | ingress-chaos-ipa-test-frontend |
| Role | role-<projekt>-<umgebung>-<berechtigung> | role-chaos-ipa-test-read |

Erläuterungen zur Namenskonvention:

* <projekt> ist der Projektname oder ein spezifisches Kürzel, das das Projekt beschreibt, in diesem Projekt dementsprechend «chaos-ipa».
* <umgebung> bezieht sich auf die Umgebung wie prod (Produktion), dev (Entwicklung), test, usw.
* <anwendung> bezieht sich auf die spezifische Anwendung, die durch die Ressource bereitgestellt wird, z.B. app (Anwendung), db (Datenbank), usw.
* <unique-id> ist eine einzigartige Kennung, die verwendet wird, um spezifische Pods zu identifizieren. Diese ID wird dabei von Kubernetes automatisch erstellt.
* <zweck> oder <berechtigung> beschreibt den spezifischen Zweck oder die Berechtigungsebene eines Secrets oder einer Rolle, z.B. credentials, read.

## Netzwerkkonzept

In diesem Abschnitt wird das Netzwerkkonzept für die erwähnte Systemumgebung definiert, damit es nicht zu Fehlkonfigurationen kommen kann. Dabei werden jeweils Services zusammen gruppiert, welche relevant für das Netzwerk sind, und darunter stehend eine Tabelle mit den jeweiligen Konfigurationen. Wichtig zu erwähnen hierbei ist, dass das Netzwerk keinem «Hochverfügbarkeits»-Prinzip folgt und somit auch nicht überall Redundanzen eingebaut werden, da es sich hierbei nur um eine Testumgebung handelt und so Ressourcen, inklusive Kosten, eingespart werden können.

### Virtual Private Cloud

|  |  |  |
| --- | --- | --- |
| Komponente | Beschreibung | Konfiguration |
| VPC | Hauptnetzwerkcontainer für Ressourcen | CIDR Block: 10.0.0.0/16 |
| Public Subnet 1 | Subnetz für öffentlich zugängliche Ressourcen | CIDR Block: 10.0.1.0/24 in AZ us-west-2a |
| Public Subnet 2 | Subnetz für öffentlich zugängliche Ressourcen | CIDR Block: 10.0.2.0/24 in AZ us-west-2b |
| Private Subnet 1 | Subnetz für interne Ressourcen | CIDR Block: 10.0.3.0/24 in AZ us-west-2a |
| Private Subnet 2 | Subnetz für interne Ressourcen | CIDR Block: 10.0.4.0/24 in AZ us-west-2b |
| Internet-Gateway | Zugangspunkt für das Internet | Angebunden an VPC |
| NAT Gateway | Ermöglicht privaten Subnetzen Internetzugang | Installiert in Public Subnet 1 |

### Elastic Kubernetes Service (EKS)

|  |  |  |
| --- | --- | --- |
| Komponente | Beschreibung | Konfiguration |
| EKS Cluster | Der Kern der Kubernetes Umgebung | Vernetzt mit VPC |
| EKS Node Groups | Gruppen von Nodes innerhalb des Clusters | Positioniert sich automatisch in privaten Subnetzen |

### Sicherheitsgruppen

|  |  |  |
| --- | --- | --- |
| Komponente | Beschreibung | Konfiguration |
| Control Plane SG | Schützt die Kubernetes-API | Erlaubt Traffic nur von vertrauenswürdigen IPs |
| Node Group SG | Schützt die Kubernetes-Nodes | Inbound von Control Plane, Outbound zum Internet |

### Load Balancer

|  |  |  |
| --- | --- | --- |
| Komponente | Beschreibung | Konfiguration |
| ALB | Verteilt eingehenden Traffic | Positioniert sich automatisch in öffentlichen Subnetzen, routet Traffic zu Services in privaten Subnetzen |

### Route Tables

|  |  |  |
| --- | --- | --- |
| Komponente | Beschreibung | Konfiguration |
| Public Route Table | Routet Traffic von öffentlichen Subnetzen | Verbindet zu Internet-Gateway |
| Private Route Table | Routet Traffic von privaten Subnetzen | Verbindet zu NAT Gateway |

## Berechtigungskonzept

In diesem Abschnitt wird das Berechtigungskonzept definiert, welches relevant für die Sicherheit des Systems ist. Da es sich dieses Projekt besonders auf den AWS EKS fokussiert, wird das Berechtigungskonzept innerhalb von AWS sehr einfach gehalten und der Fokus liegt mehr auf die Berechtigungen innerhalb des EKS. Ebenso existieren hierbei auch externe Einschränkungen, da ich keinen Full-Admin Access auf der AWS Plattform habe, da dieses Projekt innerhalb einer firmeninternen Sandbox ausgeführt wird.

### Berechtigungen innerhalb AWS

In diesem Berechtigungskonzept für AWS, setzen wir die Berechtigungen nach den Best Practices von AWS selbst um. Somit werden keine User mit spezifischen Berechtigungen erstellt, sondern es werden sogenannte Rollen erstellt, welche sich ein User selbst zuweisen kann und somit über diese Rolle seine Berechtigungen erhält. Ebenso wird hierbei auch auf das «Least Privilege» Prinzip geachtet, wonach Rollen & User nur die wichtigsten Berechtigungen erhalten.  
  
**Rollen:**Bei AWS werden normalerweise Rollen, Benutzern und Gruppen, die Berechtigungen nicht direkt zugewiesen, sondern es wird jeweils eine eigene Policy dafür erstellt. Die untenstehende Tabelle führt dabei auf, welche Rolle mit welcher Policy zusammenhängt.

|  |  |  |
| --- | --- | --- |
| Rolle | Policy | Beschreibung |
| sandbox (firmenintern vorgegeben) | sandbox | Mit dieser Rolle hat man Full-Access auf jeden einzelnen Service innerhalb von AWS. Jedoch können nur die eignene Ressourcen gesehen und bearbeitet werden. |
| chaos-ipa-read\* | policy-chaos-ipa-test-fullread | Diese Rolle erlaubt den Read-Access auf alle Ressourcen innerhalb von AWS, welche mit dem Tag «chaos-ipa» gekennzeichnet werden. |

\* Bei dieser Rolle handelt es sich um eine theoretische Rolle, welche so praktisch nicht umgesetzt werden kann, aufgrund von fehlenden Berechtigungen.  
  
**Policies:**Die nachfolgende Tabelle definiert die Policies und deren jeweiligen Berechtigungen.

|  |  |
| --- | --- |
| Policy | Berechtigungen |
| sandbox | Full-Access auf alle eigenen AWS Services |
| policy-chaos-ipa-test-fullread | Read Access auf Ressourcen mit dem Tag «chaos-ipa» |

### Berechtigungen innerhalb des EKS

In diesem Berechtigungskonzept handelt es sich um die Kubernetes RBAC (Role-Based Access Control), in welchem auch für jeweilige Rollen, diverse Berechtigungen vergeben werden. Hierbei werden ähnlich wie bei AWS, zwei verschiedene Rollen eingesetzt:

|  |  |  |
| --- | --- | --- |
| ClusterRole | Zugriffsberechtigungen | Beschreibung |
| role-chaos-ipa-test-admin | Vollzugriff auf alle Ressourcen und Operationen | Vollständige administrative Kontrolle über den gesamten Cluster. |
| role-chaos-ipa-test-read | Lesender Zugriff auf alle Ressourcen | Ermöglicht Benutzern das Ansehen von Ressourcen, aber keine Änderungen. |

Da der eigentliche Test und die Implementierung der Software von Chaos-Mesh im Vordergrund steht, sind jeweils nur Admin und Read Rollen nötig, da man entweder alles konfiguriert und erstellt oder als dritt Person sich das Tool und die Architektur ansehen möchte.

## ISDS-Konzept

Das ISDS-Konzept ist ein wichtiger Bestandteil eines jeden Projektes, da der Datenschutz und die Informationssicherheit heutzutage extrem wichtig sind. Auch in diesem Projekt wird ein ISDS-Konzept erarbeitet, obwohl es keine schützenswerten Informationen gäbe, da alles in einer Testumgebung stattfindet und ein ungewünschter Zugang zu diesen Informationen und Daten keine verehrende Folgen hätte. Folgende Daten und Informationen müssen, während diesem Projekt geschützt werden.

### Informationsschutz

Folgende Massnahmen sind einzuhalten, um entsprechende Informationen zu schützen.

|  |  |  |
| --- | --- | --- |
| Informationsschutz Nr. | Bezeichnung | Beschreibung |
| IS-1 | Klassifizierung | Alle Dokumente, welche diesem Projekt angehören, müssten korrekt klassifiziert und dementsprechend behandelt werden. |
| IS-2 | Passwörter | Es müssen stets sichere Passwörter verwendet werden. Passwörter dürfen nicht weitergegeben werden oder an unsicheren Orten gesichert werden. |
| IS-3 | Clear Desk | Es dürfen keine Dokumente mit vertraulichen Daten auf dem Schreibtisch liegen gelassen werden. |

### Datenschutz

Folgende Massnahmen sind einzuhalten, um entsprechende Daten zu schützen.

|  |  |  |
| --- | --- | --- |
| Datenschutz Nr. | Bezeichnung | Beschreibung |
| DS-1 | Personendaten | Personenbezogene Daten, welche sich in diesem Dokument befinden, dürfen nicht weiter gegeben werden |
| DS-2 | Netzplandaten | Daten wie IPs, Hostname oder ähnliches, dürfen ausserhalb dieses Projektes ebenfalls nicht weitergegeben werden. |
| DS-3 | Dokumentation | Die Klassifizierung dieser und anderen Projektdokumentation sind entsprechend zu behandeln |
| DS-4 | Prozessdaten | Daten über firmeninterne Prozesse dürfen nicht weitergegeben werden. |

### Gefahren

Zusätzlich zum Datenschutz und Informationsschutz, können stets weitere Gefahren auftreten, welche beide Kategorien gefährden können. Um diese Gefahren einzuschränken, folgt nun eine Tabelle mit möglichen Gefahren und deren Folgen:

|  |  |
| --- | --- |
| Gefahr Nr. | Gefahr |
| G01 | Konfigurationsfehler bei der Installation von Chaos-Mesh führen zu unkontrollierten Zugriffen und Datenlecks. |
| G02 | Unbefugter Zugriff auf die Terraform State-Dateien im S3-Bucket, was zu Datenmanipulation führen könnte. |
| G03 | Mangelnde Netzwerksicherheit führt zu möglichen Denial-of-Service-Angriffen auf das Kinoticketverkauf System. |
| G04 | Fehlerhafte Integration und Konfiguration des Monitoring-Tools führt zu blinden Flecken in der Überwachung. |
| G05 | Schadcode-Injektion durch Dritte in öffentlich zugängliche Repositories, die für die Pipeline verwendet werden. |

Für all diese Gefahren können auch Massnahmen entwickelt werden, folgend sind nun mögliche Massnahmen aufgelistet:

|  |  |  |
| --- | --- | --- |
| Massnahme Nr. | Gefahr Nr. | Massnahme |
| M01 | G01 | Die Installation wird gemäss offizieller Dokumentation von Chaos-Mesh oder gemässe Betriebshandbuch vorgenommen. |
| M02 | G02 | Durch den Einsatz der festgelegten Rollen und Policies, kann ein unbefugter Zugriff auf ein Minimum reduziert werden. |
| M03 | G03 | Diese Gefahr kann mittel Chaos-Mesh getestet werden und nach der Analyse verbessert werden. |
| M04 | G04 | Diese Gefahr kann mittel Chaos-Mesh getestet werden und nach der Analyse verbessert werden. |
| M05 | G05 | Implementierung von Security-Scans und Audit-Tools in den CI/CD-Pipelines, um Code-Injektionen vor der Deployment-Phase zu erkennen. |

## Testkonzept

Ein Testkonzept ist eine fundamentale Komponente, welches sicherstellt, dass ein entwickeltes System, welches innerhalb eines Projektes umgesetzt wird, auf seine Funktionalität und Härte getestet wird. Hierbei werden verschiedene Tests definiert, welche anschliessend in der Realisierungsphase umgesetzt, analysiert und ausgewertet werden. Es werden Testvoraussetzungen und Testmethoden festgelegt, sowie auch Testanforderungen und das erwartete Ergebnis.

### Testvorgehen

Die Testphase verläuft nach untenstehender Grafik.  
Ein Bild, das Text, Screenshot, Schrift, pink enthält.

Automatisch generierte Beschreibung

Abbildung 16: Testvorgehen

Der Schritt «Fehlerkorrektur» wird nur durchgeführt, falls Fehler auftreten sollten. Ansonsten wird dieser Schritt nicht durchgeführt.

### Das zu testende System & Umgebung

Das zu testende System ist eine Full-Stack-Applikation, die ein Kinoticketverkaufssystem simuliert, einschliesslich Backend, Frontend und zugehöriger Datenbankdienste. Diese Applikation wird in einer Cloud-Infrastruktur betrieben, die auf Kubernetes (EKS) basiert und mit Terraform automatisiert bereitgestellt wird. Die Umgebung beinhaltet Netzwerkelemente wie VPC, Subnetze, Internet-Gateways, NAT Gateways und Route Tables, sowie IAM-Rollen für Zugriffsberechtigungen. Die Infrastruktur und der Betriebszustand der Applikation werden mittels CloudWatch überwacht.

### Abgrenzung der Testkonzepte

Bei der Konzeption der Chaos-Engineering-Tests wird bewusst darauf verzichtet, bestimmte Szenarien zu testen. Diese Entscheidung beruht auf der Abwägung von Risikopriorität, Relevanz und den technischen Grenzen der Testumgebung:

* **Chaos-Experimente:** Es werden keine Chaos-Experimente getestet, da diese in einem separaten Teil innerhalb der Realisierung durchgeführt, analysiert und ausgewertet werden. Dies würde ansonsten zu Redundanzen führen. Jedoch zum Sicherstellen, ob die Chaos-Experimente funktionieren, wird ein einzelnes Experiment in die Testfälle aufgenommen, damit die Funktionalität des Tools sichergestellt werden kann.
* **Externe Dienste:** Ausfälle oder Beeinträchtigungen von Diensten, die ausserhalb unserer Cloud-Umgebung gehostet werden und nicht durch das Kinoticketverkaufssystem oder die Infrastruktur beeinflussbar sind, werden nicht getestet. Dazu zählen Ausfälle von Drittanbieterdiensten, überregionale Internetstörungen und Katastrophenszenarien, die eine umfassende physische Zerstörung der Infrastruktur bewirken könnten.
* **Nicht-technische Aspekte:** Aspekte wie menschliches Fehlverhalten, Betrug oder physische Sicherheit sind ebenfalls ausgeschlossen, da sie über den Rahmen und die Ziele des Chaos-Engineering hinausgehen. Der Fokus liegt auf der technischen Resilienz und der Reaktionsfähigkeit der Infrastruktur und Applikation.
* **Hardware-Ausfälle:** Das Testkonzept konzentriert sich auf softwaredefinierte Infrastruktur, wobei man davon ausgehen kann, dass die zugrundeliegende physische Hardware durch den Cloud-Dienstanbieter gewartet und bei Bedarf ersetzt wird. Spezifische Hardware-Ausfälle, wie z.B. Festplattendefekte oder Netzwerkkomponentenausfälle, sind somit nicht Teil der Testszenarien.
* **Performanztests unter extremen Bedingungen:** Obwohl während der Realisierung die beiden Experimente «CPU- und Speicherchaos» durchgeführt werden, werden extreme Belastungstests, die über die vordefinierte Kapazität der Systeme hinausgehen, nicht durchgeführt, da sie den Rahmen der normalen Betriebsbedingungen sprengen und keine realistischen Ergebnisse liefern würden.
* **Technische Grundvoraussetzungen:** In diesem Testkonzept wird vorausgesetzt, dass gewisse Grundbedingungen erfüllt sind. Somit werden Tests, welche Verfügbarkeit einer Internetverbindung oder die Kompatibilität mit allen möglichen Browser-Versionen testen, nicht durchgeführt. Man nimmt an, dass Nutzer eine stabile und angemessene Internetverbindung haben und einen aktuellen Browser verwenden. Somit fokussiert sich dieses Testkonzept auf die Resilienz des Systems unter der Annahme, dass diese grundlegenden technischen Anforderungen erfüllt sind.

### Relevante Testmittel

Die Testmittel umfassen die gesamte Cloud-Infrastruktur, die bereitgestellte Full-Stack-Applikation und die Chaos-Mesh-Komponenten. Die Hardware-Ressourcen sind virtuelle Maschinen und andere Cloud-Ressourcen. Netzwerkressourcen umfassen VPC, Subnetze und Internet/NAT Gateways. Ebenso wird auch ein üblicher Endnutzer-Client verwendet, mit welchem der Zugriff auf die Applikation getestet werden soll.

### Globale Testziele

Über alle Tests hinweg, möchte man ein globales Testziel erreichen. Das globale Testziel ist darauf ausgerichtet, die Robustheit, Zuverlässigkeit und Ausfallsicherheit der Cloud-Infrastruktur und der Kinoticketverkauf-Applikation zu gewährleisten. So soll das Vertrauen in die Infrastruktur und Applikation gestärkt werden, indem nachgewiesen wird, dass diese unter verschiedenen Bedingungen und bei verschiedenen Arten von Ausfällen und Störungen so funktionieren, wie es von ihnen erwartet wird.

### Testmethoden

Für das Erarbeiten und Durchführen der Testfälle, werden folgende Testmethoden angewendet:

* **Black-Box-Tests:** Diese Tests überprüfen die Verfügbarkeit und Funktionalität der Applikation aus Nutzersicht, ohne Kenntnis des internen Codes.
* **White-Box-Tests:** Hier wird das interne Verhalten der Applikation getestet, wobei Kenntnisse über den Code genutzt werden, um die interne Logik und Prozesse zu überprüfen.
* **Chaos-Engineering:** Anwendung von geplanten, kontrollierten Experimenten, um die Stabilität und Fehlertoleranz der Infrastruktur und Applikation unter Stressbedingungen zu testen.

Durch diese Testmoden kann sichergestellt werden, dass sowohl die Benutzererfahrung, sowie auch die technische Integrität und Belastbarkeit des Systems sichergestellt werden kann.

### Mängelklassifizierung

Die jeweiligen Testergebnisse, werden diversen Fehlerklassen zugeteilt, um festzustellen, ob anhand der Testergebnisse, weitere Massnahmen getroffen werden müssen.

|  |  |  |  |
| --- | --- | --- | --- |
| Fehlerklassen ID | Ausmass | Beschreibung | |
| FK00 | Mangelfrei | Der Test verlief problemlos. Es müssen keine Massnahmen getroffen werden | |
| FK01 | Unwesentlicher Mangel | | Es trat ein kleiner Mangel auf, das System kann weiterhin verwendet werden. |
| FK02 | Wesentlicher Mangel | | Der Mangel beeinträchtigt die Funktionalität. Das System kann jedoch weiterhin verwendet werden. |
| FK03 | Kritischer Mangel | | Die Anforderungen wurden nicht erfüllt. Der Mangel muss schnellstmöglich behoben werden. |

Sollte eine Fehlerklasse über der Klasse «FK00» liegen, so müssen alle Mängel dokumentiert werden.

### Testfälle

Folgend werden die Testfälle definiert und jeweils tabellarisch dargestellt.

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF01 | Bereitstellung der Infrastruktur mit Terraform |

|  |
| --- |
| Anforderungen |
| Eine Internetverbindung muss gewährleistet sein, sowie der Zugang über die AWS CLI. Alle Scripts müssen vorhanden und komplett sein. |

|  |
| --- |
| Vorgehen |
| 1. Mit der AWS CLI verbinden. 2. «terraform apply» in der Konsole eintippen. 3. Warten, bis die Infrastruktur bereitgestellt ist. 4. Sicherstellen, dass alles korrekt bereitgestellt wurde. |

|  |
| --- |
| Erwartetes Ergebnis |
| Die Infrastruktur wird vollständig und fehlerfrei aufgesetzt, allein durch Terraform. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF02 | Frontend ist erreichbar und nutzbar |

|  |
| --- |
| Anforderungen |
| Eine Internetverbindung muss gewährleistet sein. Ebenso muss die Infrastruktur bereits bereitgestellt sein. |

|  |
| --- |
| Vorgehen |
| 1. Mit dem Frontend verbinden 2. Kino-Film auswählen 3. Sitzplatz auswählen 4. Bezahlprozess abschliessen 5. Prüfen ob Sitze erfolgreich reserviert wurden |

|  |
| --- |
| Erwartetes Ergebnis |
| Das Frontend ist funktional und es können alle Funktionen verwendet werden. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF03 | Chaos-Mesh ist installiert und zugriffsbereit |

|  |
| --- |
| Anforderungen |
| Die Infrastruktur muss vollständig bereitstehen, zudem muss eine Verbindung zu Kubernetes bestehen. |

|  |
| --- |
| Vorgehen |
| 1. Prüfen, ob Chaos-Mesh in die Kubernetes Umgebung integriert wurde. 2. Prüfen, ob eine Verbindung auf das Chaos-Mesh Dashboard möglich ist. |

|  |
| --- |
| Erwartetes Ergebnis |
| Chaos-Mesh ist einsatzbereit und es kann zugegriffen werden. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF04 | Monitoring über CloudWatch funktioniert |

|  |
| --- |
| Anforderungen |
| Eine Internetverbindung muss gewährleistet sein. Ebenso müssen die Infrastruktur und die Applikation bereits bereitgestellt sein. |

|  |
| --- |
| Vorgehen |
| 1. Mit der AWS Konsole verbinden. 2. EKS aufsuchen. 3. Auf den Reiter «CloudWatch» navigieren. 4. Prüfen, ob Metriken der Systemumgebung angezeigt werden. |

|  |
| --- |
| Erwartetes Ergebnis |
| Im CloudWatch Dashboard sind aktuelle Metriken erkennbar in Bezug auf die Infrastruktur. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF05 | GitHub Actions Pipeline ist funktionstüchtig |

|  |
| --- |
| Anforderungen |
| Infrastruktur, Applikation und GitHub Actions Pipeline muss bereitgestellt sein. |

|  |
| --- |
| Vorgehen |
| 1. Änderungen an einem beliebigen Code vornehmen. 2. Einen Commit und Push über GitHub vornehmen. 3. Prüfen, ob die Pipeline Änderungen erkennt und ein Re-Deployment vornimmt. |

|  |
| --- |
| Erwartetes Ergebnis |
| Die Pipeline erkennt ob Änderungen vorgenommen wurden, und reagiert entsprechend auf die Änderungen mit einem Re-Deployment. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF06 | Einsicht des Terraform-States auf einem anderen Gerät |

|  |
| --- |
| Anforderungen |
| Infrastruktur und Applikation muss bereitgestellt sein und aktueller Terraform State muss in einem S3 Bucket gesichert sein. |

|  |
| --- |
| Vorgehen |
| 1. Klonen des GitHub Repository auf eine andere Instanz 2. Nötige Credentials setzen 3. Terraform Plan ausführen 4. Prüfen, ob der aktuelle Terraform State abgerufen werden kann. |

|  |
| --- |
| Erwartetes Ergebnis |
| Mit Terraform Plan wird eine Auskunft über das System gegeben. Dabei sollte sich der State nicht unterscheiden mit dem State, welcher im S3 Bucket gesichert ist, da dieser State per Remote gekoppelt ist. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF07 | Chaos-Mesh Experiment funktioniert |

|  |
| --- |
| Anforderungen |
| Infrastruktur, Applikation und Chaos-Mesh muss bereitgestellt sein. |

|  |
| --- |
| Vorgehen |
| 1. Verbinden auf das Chaos-Mesh Dashboard. 2. Ausführen eines belieben Experimentes. 3. Prüfen, ob Experiment erfolgreich auf die Applikation angewendet wurden konnte. |

|  |
| --- |
| Erwartetes Ergebnis |
| Es kann ein beliebiges Experiment auf die Applikation ausgeführt werden. Nach Abschluss des Experimentes, ist dies im Chaos-Mesh Dashboard ersichtlich. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

|  |  |
| --- | --- |
| Testfall Nr. | Testfall |
| TF08 | Installation von Chaos-Mesh erfolgt nach Benutzeranleitung |

|  |
| --- |
| Anforderungen |
| Infrastruktur und Applikation muss bereitgestellt sein. |

|  |
| --- |
| Vorgehen |
| 1. Zugriff auf die AWS CLI & Kubernetes sicherstellen 2. Installation von Chaos-Mesh über das Benutzerhandbuch, ohne externe Hilfe. 3. Prüfen, ob Chaos-Mesh erfolgreich erstellt wurde. |

|  |
| --- |
| Erwartetes Ergebnis |
| Ein Benutzer kann Chaos-Mesh in eine vorhandene Kuberenetesumgebung integrieren, dies anhand der Benutzeranleitung. |

|  |  |
| --- | --- |
| Eingetroffenes Ergebnis | Fehlerklasse |
|  |  |

|  |
| --- |
| Durchführung und Freigabe |
| Datum, Name |

## Automatisierungskonzept

Das Automatisierungskonzept stellt sicher, dass bei Änderungen an der Codebase, automatische Massnahmen getroffen werden. Ziel dabei ist es, dass schlussendlich mit Chaos-Mesh ein Experiment automatisch durchgeführt werden kann. So kann bei Änderungen an der Infrastruktur oder der Applikation getestet werden, ob diese weiterhin Widerstandsfähig sei, oder ob weitere Anpassungen gemacht werden müssen. Dieses Konzept könnten man anschliessend, in einer echten Umgebung, auf viele verschiedene Experimente anwenden.  
  
Folgend nun ein Diagramm, welches die Automatisierung visuell in einem Flussdiagramm darstellen soll.  
  
Ein Bild, das Text, Visitenkarte, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 17: Flussdiagramm Automatisierungskonzept

In diesem Flussdiagramm ist ersichtlich, dass die Automatisierung verschiedene Aktionen trifft, basierend auf dem jeweiligen Repository, welche Änderungen erhält. So muss nicht jedes Mal die komplette Infrastruktur neu bereitgestellt werden. Dies hat zur Folge, dass so Ressourcen und Zeit eingespart werden können, da eine vollständige Bereitstellung der Infrastruktur jeweils über 20 Minuten dauern kann.  
  
Somit: Wenn Änderungen an den Terraform-Scripts erkannt werden, wird alles vollständig neu bereitgestellt. Sollten nur Änderungen an der Applikation vorgenommen werden, so werden nur neue Images generiert und diese ersten anschliessend die aktuellen Container in der Kubernetesumgebung.  
  
So kann eine effektive Automatisierung der Umgebung und gleichzeitiges Chaos-Testing gewährleistet werden.

## Phasenfreigabe Realisierung

Durch die Unterschrift des Auftraggebers und des Projektleiters wird festgelegt, dass die Konzeptphase abgeschlossen wurde und somit die Realisierungsphase freigegeben ist.

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Name | Rolle | Unterschrift |
| 26.04.2024 | Justin Winistörfer | Projektleiter |  |
| 26.04.2024 | Mathias Hirth | Auftraggeber |  |

# Realisierung

Nach der vorherigen Phasenfreigabe startet, laut HERMES 5.1, die Realisierungsphase des Projektes. In dieser Phase geht es nun darum, das konzipierte Produkt umzusetzen. Sobald das Projekt danach umgesetzt wurde, werden anhand des Testkonzeptes, die Tests durchgeführt und dokumentiert. Sobald diese Phase durchlaufen ist, und eine Phasenfreigabe stattgefunden hat, wird mit der Einführung fortgefahren.



Abbildung 18: Realisierungsphase

## Bereitstellung der Infrastruktur mit Terraform

Die gesamte Infrastruktur, welche für dieses Projekt nötig ist, wird wie schon erwähnt, auf der Cloud Plattform AWS bereitgestellt. Damit man jedoch die Möglichkeit hat, diese Projektinfrastruktur nach Bedarf hoch oder runterzufahren, um Kosten und Ressourcen einzusparen, wird die gesamte Plattform mit Terraform bereitgestellt.

### Terraform Erklärung

Terraform ist ein mächtiges Tool, welches ermöglicht gesamte Infrastrukturen über einen oder mehrere Scripts bereitzustellen. So wird dieses Vorgehen meist unter IaC (Infrastructure as Code) bezeichnet. Ganz einfach erklärt ist Terraform folgendes:  
  
„Terraform ist ein Werkzeug, das IT-Spezialisten dabei hilft, Computerressourcen automatisch zu verwalten und einzurichten. Stellen Sie sich vor, Sie möchten eine Reihe von LEGO-Bauten nach einer Anleitung automatisch erstellen lassen, anstatt jedes Stück einzeln von Hand zusammenzufügen. Terraform erlaubt es, solche Anleitungen zu schreiben, die genau festlegen, welche Teile der Computerinfrastruktur wie Server, Speicher und Netzwerke aufgebaut werden sollen, und führt diese Anleitungen dann automatisch aus, um alles genau nach Plan zu erstellen. Dies spart Zeit und verringert die Möglichkeit von Fehlern, die passieren könnten, wenn alles manuell gemacht werden würde.“[[8]](#footnote-8)  
  
Diese Erklärung wurde hierbei von ChatGPT generiert und erklärt ziemlich einfach, was Terraform ist. Somit ist Terraform genau die richtige Lösung für dieses Projekt.

### Einrichten der Entwicklungsumgebung

Damit diese Scripts zur Bereitstellung von Terraform entwickelt werden können, muss zuerst eine Entwicklungsumgebung erstellt werden. Dies ist dabei recht simple.

Folgende Sachen werden benötigt:

* Eine IDE (Integrated Development Environment) in diesem Fall VSCode
* Einen beliebigen Ordner als Speicherort
* Ein Versionierungssystem wie GitHub (optional, aber empfehlenswert)

Hat man sich einen Ordner in einer IDE erstellt und somit einen Bereich, in welchem man nun scripten möchte, so muss man nun Terraform installieren. Je nach Betriebssystem ist die Installation von Terraform anders. Hier ist ein Link, welcher erklärt, wie Terraform installiert werden kann: <https://developer.hashicorp.com/terraform/install>.

Sobald Terraform installiert wurde, kann man die Installation mit dem Befehl «terraform -v» testen. Dies alles zusammen, kann dann so aussehen:  
  
Ein Bild, das Text, Screenshot, Software, Multimedia-Software enthält.

Automatisch generierte Beschreibung

Abbildung 19: Übersicht über die Entwicklungsumgebung

Auf diesem Screenshot sieht man nun die Entwicklungsumgebung, welche für dieses Projekt eingerichtet wurde, ebenso sieht man die Terraform-Version, welche installiert wurde. Ebenso ist auch ersichtlich, dass schon ein GitHub Repository mit der Umgebung verbunden ist. Hierbei handelt es sich um das Repository, welches auch als Dokumentenablage genutzt wird.

### Zugang zu AWS

Damit Terraform die Infrastruktur automatisch auf AWS deployen kann, benötigt Terraform den Zugang auf den AWS Account. Hierfür kann man Terraform über die AWS CLI automatisch zugriff auf den eingeloggten Account geben. Dies bedeutet, dass wenn Terraform ein Deployment vornimmt, dass sich Terraform über die AWS CLI verbindet. Jedoch ist es zuerst wichtig, dass man in der AWS CLI eingeloggt ist.  
  
Loggt man sich jedoch in die CLI ein, so verfällt das Login jede Stunde, aufgrund von firmeninternen Vorgaben. Dies ist besonders mühsam, da der Log In-Prozess über die AWS CLI ziemlich aufwändig ist, und zusätzlich noch eine Multi-Factor-Authentication (MFA) verlangt wird. Damit dieser aufwändige Log In Prozess nicht jedes Mal durchgeführt werden muss, wurde dafür ein Shell-Script erstellt, welches diese Arbeit abnimmt. Dieses Shell-Script ist im Anhang der IPA hinterlegt.

So kann man nun jederzeit, wenn man sich in die CLI einloggen muss, einfach nur das Shell-Script ausführen, danach wird man nach dem MFA-Code gefragt, welcher man angeben muss, und schon ist man mit der AWS CLI angemeldet. Dies sieht dann so aus:

Ein Bild, das Text, Screenshot, Schrift enthält.

Automatisch generierte Beschreibung

Abbildung 20: Log-In in die AWS CLI

Anhand des Exports sieht man, dass erfolgreich die nötigen Keys und Tokens gesetzt wurde. Aufgrund der Sicherheit wurden die jeweiligen Keys und Tokens zensiert. Nun kann Terraform so auch schon mit AWS kommunizieren und die Infrastruktur später bereitstellen. Wichtig zu erwähnen ist, dass dieses Script jede Stunde ausgeführt werden muss.

### Bereitstellung der AWS Services mit Terraform

Sobald die Entwicklungsumgebung erstellt wurde und der Zugang zu AWS sichergestellt wurde, kann man nun beginnen die Scripts, für das Bereitstellen der Infrastruktur durch Terraform, zu erstellen.

Folgend wird auf die einzelnen Scripts und deren Funktionalität eingegangen:

Abbildungsverzeichnis

[Abbildung 1: Schnellzugriffspalette 9](#_Toc51839769)

[Abbildung 2: Eine Beispielabbildung 10](#_Toc51839770)

[Abbildung 3: Eine Beispieltabelle 11](#_Toc51839771)

[Abbildung 4: Tabellenformatoptionen 11](#_Toc51839772)

Tabellenverzeichnis

[Tabelle 1: Textformate Überschriften 8](#_Toc15489977)

[Tabelle 2: Textformate Fließtext 8](#_Toc15489978)

[Tabelle 4: Textformate Tabellen 8](#_Toc15489979)

[Tabelle 5: Textformate Beschriftungen/Verweise 9](#_Toc15489980)

[Tabelle 6: Textformate Sonstiges 9](#_Toc15489981)

# Einleitung

Dieses Dokument dient Ihnen als Word-Dateivorlage

In dieser Datei sind die gängigsten Formatvorlagen definiert, mit denen Sie die Texte schnell gliedern und gestalten können.

## Dokument als Standardvorlage nutzen

Abspeichern als Word-Vorlage mit Makros unter   
#appData#\Roaming\Microsoft\Templates

mit dem Dateinamen „Normal.dotm“[[9]](#footnote-9)

## Übersicht über die wichtigsten Textformatierungen

|  |  |  |
| --- | --- | --- |
| Überschriften | Anmerkungen | AbsatzFormat/Textformat |
| 1 Überschrift 1 |  | Absatz |
| 1.1 Überschrift 2 |  | Absatz |
| 1.1.1 Überschrift 3 |  | Absatz |
| 1.1.1.1 Überschrift 4 |  | Absatz |
| Absatzüberschrift | Nicht nummeriert | Absatz |

Tabelle 1: Textformate Überschriften

|  |  |  |
| --- | --- | --- |
| Fließtext | Anmerkungen | AbsatzFormat/Textformat |
| Fließtext |  | Absatz |
| Standard | Basisformat für Text | Absatz |
| Fett | Fettstellung bitte nicht über die „F“ Taste | Text |
| Liste mit Bullets | Unterpunkte in gesonderten Formatvorlagen | Absatz |
| Liste sortiert | Unterpunkte in gesonderten Formatvorlagen | Absatz |
| Hyperlink |  | Text |
| Besuchter Link |  | Text |
| Zitate |  | Absatz |

Tabelle 2: Textformate Fließtext

|  |  |  |
| --- | --- | --- |
| Tabellen | Anmerkungen | AbsatzFormat/Textformat |
| Text Tabelle | Links ausgerichtete Inhalte (meist Text) | Absatz |
| Tabelle Zahl | Rechts ausgerichtete Inhalte (meist Zahlen) | Absatz |
| Tabellenkopf Text | Kopfzeile passend zum Inhalt darunter | Absatz |
| Tabellenkopf Zahl | Kopfzeile passend zum Inhalt darunter | Absatz |

Tabelle 3: Textformate Tabellen

|  |  |  |
| --- | --- | --- |
| Beschriftungen/Verweise | Anmerkungen | AbsatzFormat/Textformat |
| Beschriftung | für Abbildungen/Tabellen | Absatz |
| Objekt Abbildungen | Setzt eingefügte Abbildung zentriert und mit korrekten Abständen | Absatz |
| Fußnotentext |  | Absatz/Text |
| Fußnotenzeichen |  | Text |
| Verzeichnis im IHV | Kapitel, die nicht Bestandteil des Dokumentinhalts sind (also nicht nummeriert werden) und im Inhaltsverzeichnis aufgeführt werden | Absatz |
| Verzeichnis nicht im IHV | Kapitel, die nicht Bestandteil des Dokumentinhalts sind (also nicht nummeriert werden) und auch nicht im Inhaltsverzeichnis aufgeführt werden | Absatz |
| Verzeichnis 1 | Verzeichnis im IHV und Kapitel im Inhaltsverzeichnis | Absatz |
| Verzeichnis 2 | 1.1 Überschrift 2 im Inhaltsverzeichnis | Absatz |
| Verzeichnis 3 | 1.1.1 Überschrift 3 im Inhaltsverzeichnis | Absatz |

Tabelle 4: Textformate Beschriftungen/Verweise

|  |  |  |
| --- | --- | --- |
| Sonstiges | Anmerkungen | AbsatzFormat/Textformat |
| Fußzeile |  | Absatz |
| Kopfzeile |  | Absatz |

Tabelle 5: Textformate Sonstiges

## Formatierung von Texten

Wählen Sie die Formate über den Formtvorlagenkatalog (Strg + Alt + Umschalt + S)  
oder über den Schnellzugriff oben in der Menüzeile aus

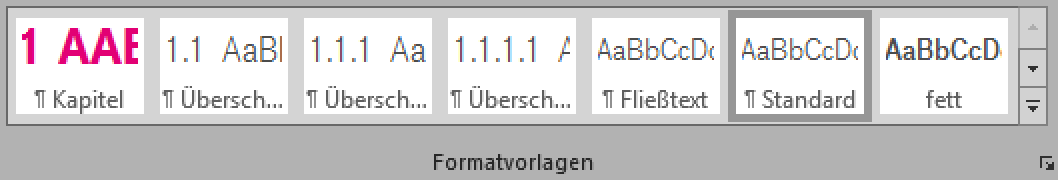


Abbildung 21: Schnellzugriffspalette

# Der Umgang mit Bildern und Tabellen

## Platzierung von Bildern

Dieser Absatz ist in „Standard“ formatiert. Der folgende Absatz (in dem das Bild eingefügt wurde) ist im Format „Objekt Abbildung“ formatiert. Damit ist das Bild direkt zentriert ausgerichtet und hat die richtigen Abstände zum Text

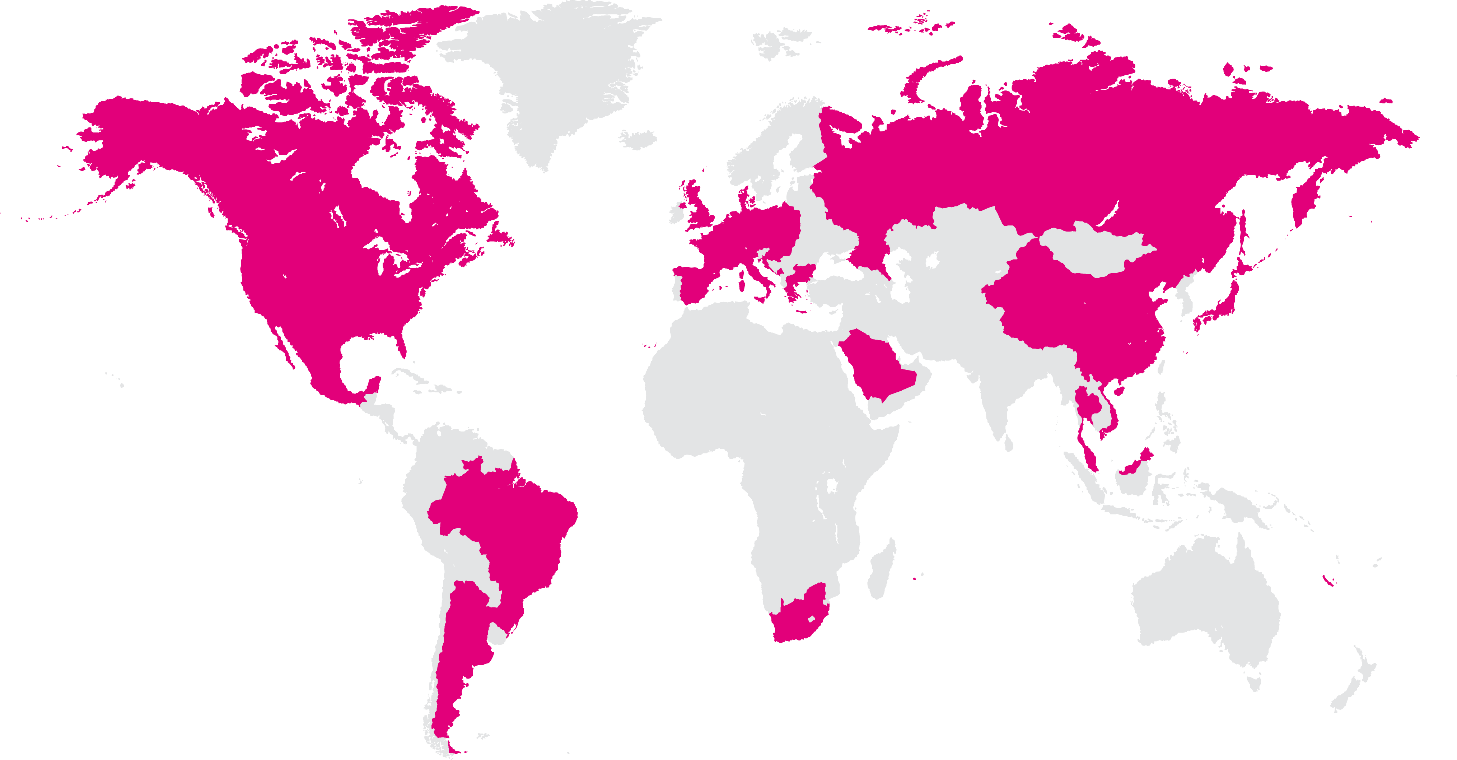


Abbildung 22: Eine Beispielabbildung

## Platzierung und Formatierung von Tabellen

So erzeugen Sie eine neue Tabelle:

1. Klicken Sie mit dem Cursor auf die Stelle, an der die Tabelle eingesetzt werden soll
2. unter „Einfügen“ > „Tabelle“ legen Sie eine Tabelle in der gewünschten Größe fest  
   Standardmäßig werden Tabellen im Texttabellen-Format angelegt  
   Handelt es sich um eine Tabelle mit Zahlen wählen Sie einfach die 2. Vorlage aus:



1. befüllen Sie die Tabelle mit Texten und Werten
2. Formatieren Sie die Texte mit den dafür vorgesehenen Formatvorlagen:

|  |  |  |
| --- | --- | --- |
|  | TabelleNKopf Text | Tabellenkopf Zahl |
| Zeile 1 | Tabelle Text. | Tabelle Zahl |
| Zeile 2 | Dies ist ein Beispiel für eine Zelle mit mehreren Absätzen.  Hier ist der zweite Absatz. | 12 345,77 |
| Zeile 3 | Eine einzeilige Zeile. | 12 345,88 |

Abbildung 23: Eine Beispieltabelle

Über die Tabellenformatoptionsfelder können Sie z.B. Kopfzeilen oder die Fettstellung der ersten Spalte an- oder ausschalten.

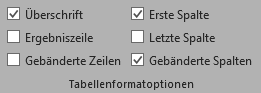


Abbildung 24: Tabellenformatoptionen

## 

# Aufzählungen

Im Folgenden werden zur Demonstration der entsprechenden Formatvorlagen einige Beispiele für unterschiedliche Arten von Aufzählungen gegeben. Sie können gelöscht werden.

## Punktaufzählungen

* Ein Absatz mit dem Format „Liste mit Bullet 1“.  
  Und hier noch eine zweite Zeile in diesem Absatz.  
  Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste mit Bullet 1 Fortsetzung“

* Ein weiterer Absatz mit dem Format „Liste mit Bullet 1“.
  + Ein Absatz mit dem Format „Liste mit Bullet 2“  
    Und hier noch eine zweite Zeile in diesem Absatz.  
    Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste mit Bullet 2 Fortsetzung“

* + Ein weiterer Absatz mit dem Format „Liste mit Bullet 2“.
    - Ein Absatz mit dem Format „Liste mit Bullet 3“.  
      Und hier noch eine zweite Zeile in diesem Absatz.  
      Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste mit Bullet 3 Fortsetzung“

* + - Ein weiterer Absatz mit dem Format „Liste mit Bullet 3“.
      * Ein Absatz mit dem „Liste mit Bullet 4“.  
        Und hier noch eine zweite Zeile in diesem Absatz.  
        Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste mit Bullet 4 Fortsetzung“

* + - * Ein weiterer Absatz mit dem Format „Liste mit Bullet 4“.
        + Ein Absatz mit dem Format „Liste mit Bullet 5“.

Ein Absatz mit dem Format „Liste mit Bullet 5 Fortsetzung“.

* + - * + Ein weiterer Absatz mit dem Format „Liste mit Bullet 5“.

## Aufzählungen mit Nummerierung

1. Ein Absatz mit dem Format „Liste sort 1“.  
   Und hier noch eine zweite Zeile in diesem Absatz.  
   Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste sort 1 Fortsetzung“.

* 1. Ein Absatz mit dem Format „Liste sort 2“.  
     Und hier noch eine zweite Zeile in diesem Absatz.  
     Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste sort 2 Fortsetzung“.

* 1. Ein weiterer Absatz mit dem Format „Liste sort 2“
     1. Ein Absatz mit dem Format Liste sort 3.  
        Und hier noch eine zweite Zeile in diesem Absatz.  
        Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste sort 3 Fortsetzung“.

* + 1. Ein weiterer Absatz mit dem Format „Liste sort 3“
    2. …
       1. Ein Absatz mit dem Format „Liste sort 4“.  
          Und hier noch eine zweite Zeile in diesem Absatz.  
          Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste sort 4 Fortsetzung“.

* + - 1. Ein weiterer Absatz mit dem Format „Liste sort 4“
      2. Ein Absatz mit dem Format Liste sort 4.
         1. Ein Absatz mit dem Format „Liste sort 5“.  
            Und hier noch eine zweite Zeile in diesem Absatz.  
            Neue Zeilen innerhalb dieses Absatzes erzeugen Sie mit „Shift“ + „Enter“

Einen neuen Absatz innerhalb dieser Liste erzeugen Sie mit dem Format   
„Liste sort 5 Fortsetzung“.

Ein weiterer Absatz mit dem Format „Liste sort 5“

### Neubeginn einer Nummerierung

Soll die Nummerierung einer Aufzählung nicht fortgesetzt, sondern neu begonnen werden, so muss dies durch die Abkopplung von der Liste erfolgen.

Ein Absatz mit dem Format Liste sort 1, dessen Nummerierung von vorn beginnen soll, wird über das Pull-Down-Menü, das durch rechten Mausklick auf die Aufzählungsnummer erscheint, entsprechend angepasst.

# Beispielanhang

Mitgeltende Unterlagen

Abkürzungsverzeichnis/Glossar

Expertenbesuchsprotokoll 23.04.2024

Anwesend: Alex Krämer, Mathias Hirth, Justin Winistörfer

Abwesend: Kevin Waldspurger

Protokollführer: Justin Winistörfer

Standort: T-Systems Schweiz AG, Zollikofen

Ablauf:

* Vorstellungsrunde
* Aufklärung für was Experten da sind
* Informationen bezüglich KI
* Erklärung der IPA
* Blick auf den Zeitplan
* Sicherstellen das IPA gewährleistet werden kann
* Abgabetermin klarstellen
* Fragen / Antworten zu den Kriterien
* Informationen zu Testkonzept
* Betrachtung des Journals
* Tipps und Tricks vom Experten
* Klären von weiteren Fragen

|  |  |
| --- | --- |
| Protokollierung | Betroffene Personen |
| Vorstellungsrunde  Die betroffenen Personen stellen sich jeweils vor. | Justin Winistörfer  Mathias Hirth  Alex Krämer |
| Aufklärung für was die Experten da sind  Es wird geklärt, dass die Experten nicht nur fürs bewerten da sind, sondern auch zum sicherstellen, dass der Kandidat die IPA besteht. | Alle |
| Informationen bezüglich KI  Herr Krämer erwähnt, dass die Nutzung von KI nicht verboten sei, jedoch IMMER gekennzeichnet werden muss. | Alex Krämer  Justin Winistörfer |
| Erklärung der IPA  Der Kanditat erklärt worum es in der IPA genau geht. | Justin Winistörfer |
| Blick auf den Zeitplan  Herr Krämer wirft einen Blick auf den Zeitplan des Kandidaten. Er beobachtet dabei, dass die SOLL Arbeitszeit pro Tag nicht 8h pro Tag und 80h Gesamthaft betragen. Ebenso ist noch ein grösserer Zeitblock vorhanden. Der Experte bitten den Kandidaten den jetzigen Stand auf PkOrg hochzuladen und den Zeitplan zu überarbeiten. Der neue Zeitplan sollte ebenso bis am 24.04.2024 um 12:00 Uhr auf PkOrg hochgeladen werden. | Alex Krämer  Justin Winistörfer |
| Sicherstellen das IPA gewährleistet werden kann  Herr Krämer stellt sicher, dass der Kandidat seine volle Zeit für die IPA nutzen kann. Ebenso wird sichergestellt, dass der Kandidat alle Dokumente und Materialien hat, welche für die Duchführung der IPA nötig sind. | Alex Krämer  Justin Winistörfer |
| Abgabetermin klarstellen  Herr Krämer erwähnt expilizit, dass die IPA genau um 18:00 Uhr am 7.5.2024 hochgeladen werden muss. Jede Sekunde zu spät = ½ Note Abzug. Ebenso wird empfohlen die IPA mehrmals hochzuladen, jedoch nicht zu signieren! | Alex Krämer |
| Fragen / Antworten zu den Kriterein  Herr Krämer stellt dem Kandidat verschiedene Fragen zu den Kritieren, um sicher zu gehen, dass dieser die Kriterien verstanden hat. | Alex Krämer  Justin Winistörfer |
| Informationen zum Testkonzept  Herr Krämer gibt wichtige Informationen und Tipps bezüglich dem Testkonzept. | Alex Krämer |
| Betrachtung des Journals  Herr Krämer wirft einen Blick auf das Lernjournal des Kandidaten, und gibt diesen als Tipps mit, dass er möglichst seine Experten im Journal mitnehmen und abholen soll. | Alex Krämer  Justin Winistörfer |
| Tipps und Tricks vom Experten  Herr Krämer gibt noch diverse wertvolle Tipps bezüglich der IPA. | Alex Krämer |
| Klären von weiteren Fragen  Der Kandidat stellt noch eigene Fragen, welche vom Experten beantwortet werden. | Justin Winistörfer |

Quellenverzeichnis

Die folgenden Einträge sind nur Beispiele, sie können gelöscht werden.

[Word MVP Site] Microsoft Word MVP FAQ Site. <http://www.mvps.org/word>

[Nicol & Albrecht 2002] Natascha Nicol und Ralf Albrecht. Wissenschaftliche Arbeiten schreiben mit Word, Formvollendete und normgerechte Examens-, Diplom- und Doktorarbeiten. Addison-Wesley, München 2002.

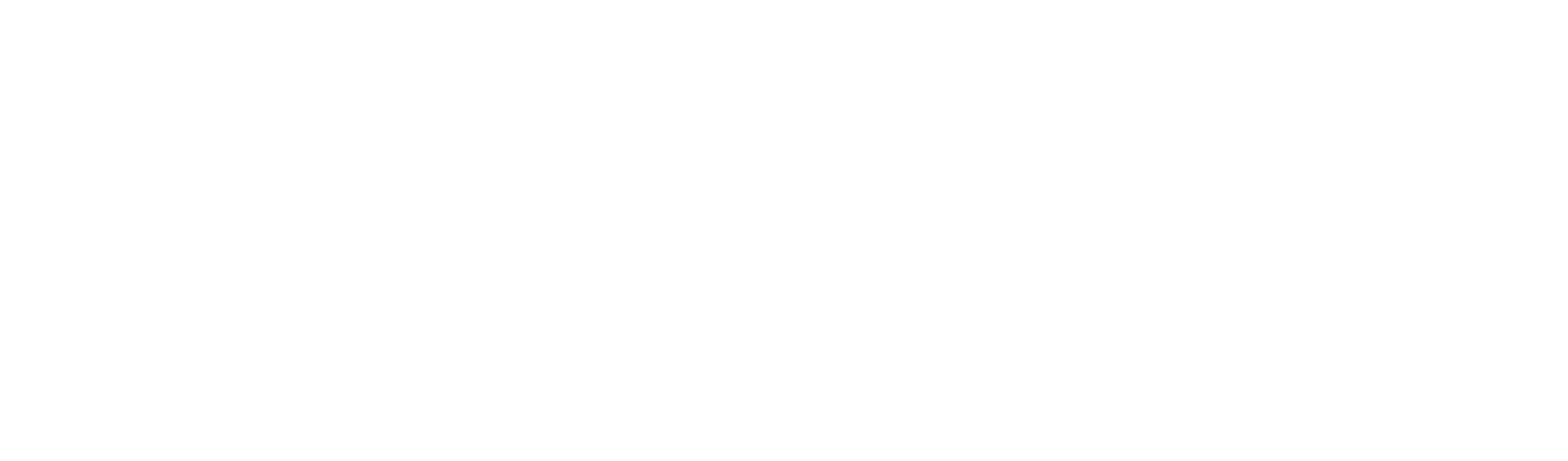
[Roman 1998] Steven Roman. Learning Word Programming. O’Reilly, Sebastopol, CA 1998.

[Weßelmann et al. 2002] Bettina Weßelmann, Johannes Wiele und Markus Tietlow. Microsoft Word 2002 – Das Handbuch. Microsoft Press, Unterschleißheim 2001.

[Willberg & Forssmann 2000] Hans Peter Willberg und Friedrich Forssmann. Erste Hilfe in Typographie, Ratgeber für den Umgang mit Schrift. Verlag Hermann Schmidt, Mainz 2000.

Expertenbesuchprotokoll 10.04.2024

Anwesend: Justin Winistörfer, Mathias Hirth, Alex Krämer



|  |  |  |
| --- | --- | --- |
| Kontakt Telekom  Straße  Stadt | Telefon: +xx xxx xxx xxx | Internet: www.  www. |

1. HERMES Methodenübersich: https://www.hermes.admin.ch/de/projektmanagement/verstehen/ubersicht-hermes/methodenubersicht.html [↑](#footnote-ref-1)
2. HERMES Szenarien: https://www.hermes.admin.ch/de/projektmanagement/verstehen/szenarien.html [↑](#footnote-ref-2)
3. HERMES Module: https://www.hermes.admin.ch/de/projektmanagement/verstehen/module.html [↑](#footnote-ref-3)
4. HERMES Phasen und Meilensteine: https://www.hermes.admin.ch/de/projektmanagement/verstehen/phasen-und-meilensteine.html [↑](#footnote-ref-4)
5. HERMES Rollen: https://www.hermes.admin.ch/de/projektmanagement/verstehen/rollen/auftraggeber.html [↑](#footnote-ref-5)
6. PkOrg Beteiligte Personen: https://2024.pkorg.ch/overview [↑](#footnote-ref-6)
7. SMART: https://de.wikipedia.org/wiki/SMART\_(Projektmanagement) [↑](#footnote-ref-7)
8. ChatGPT: <https://chat.openai.com/> [↑](#footnote-ref-8)
9. Dateivorlage mit Makros [↑](#footnote-ref-9)