**📌** Product Backlog für CMB-Prod

# Epic 1: Aufbau der IT-Infrastruktur

* **User Story 1.1:** Als Systemadministrator möchte ich **Pi’s bereitstellen**, um eine Serverumgebung zu gewährleisten.
  + ✅ Akzeptanzkriterien: Pi‘s sind herausgesucht.
  + 🛠 Tasks: Pi’s mit genug Leistung heraussuchen
* **User Story 1.2:** Als Systemadministrator möchte ich **RaspiOS auf allen Pi‘s installieren und konfigurieren**, um ein stabiles und sicheres System zu haben.
  + ✅ Akzeptanzkriterien: RaspiOS ist installiert, Netzwerkkonfiguration abgeschlossen.
  + 🛠 Tasks: Installation von RaspiOS, Netzwerkeinstellungen, Sicherheitsupdates.
* **User Story 1.3:** Als Administrator möchte ich sicherstellen, dass **alle Pi‘s miteinander kommunizieren können**, um die spätere Infrastruktur aufzubauen.
  + ✅ Akzeptanzkriterien: Alle Pi’s können sich gegenseitig erreichen.
  + 🛠 Tasks: Netzwerktests (Ping, SSH-Verbindungen).

# Epic 2: Aufbau des Webserver-Clusters

* **User Story 2.1:** Als Projektteam möchten wir **zwei Webserver bereitstellen**, die Inhalte ausliefern, um eine redundante Infrastruktur zu haben.
  + ✅ Akzeptanzkriterien: Webserver sind installiert, Testseite wird ausgeliefert.
  + 🛠 Tasks: Installation von Apache/Nginx, Bereitstellung einer Test-Webseite.
* **User Story 2.2:** Als Administrator möchte ich **einen Load Balancer einrichten**, um die Last auf die Webserver effizient zu verteilen.
  + ✅ Akzeptanzkriterien: Load Balancer ist installiert, Anfragen werden verteilt.
  + 🛠 Tasks: Installation von Nginx als Load Balancer, Konfiguration der Load-Balancing-Strategie.
* **User Story 2.3:** Als Administrator möchte ich die **Least-Connection-Strategie aktivieren**, um eine gleichmäßige Verteilung der Anfragen sicherzustellen.
  + ✅ Akzeptanzkriterien: Server mit geringerer Last erhalten neue Anfragen bevorzugt.
  + 🛠 Tasks: Anpassung der Nginx-Konfiguration, Lasttest mit simulierten Anfragen.
* **User Story 2.4:** Als Projektteam möchten wir sicherstellen, dass **defekte Webserver automatisch aus dem Load Balancer entfernt werden**, um die Verfügbarkeit zu erhöhen.
  + ✅ Akzeptanzkriterien: Load Balancer erkennt nicht erreichbare Webserver und entfernt sie.
  + 🛠 Tasks: Konfiguration von Health Checks für den Load Balancer, Test mit Serverausfällen.

# Epic 3: Bereitstellung einer dynamischen Website

* **User Story 3.1:** Als Endbenutzer möchte ich **eine dynamische Website aufrufen können**, die von den Webservern bereitgestellt wird.
  + ✅ Akzeptanzkriterien: Website ist erreichbar und funktioniert korrekt.
  + 🛠 Tasks: Entwicklung der Website (Frontend & Backend), Datenbankanbindung (falls erforderlich).

# Epic 4: Monitoring der Infrastruktur

* **User Story 4.1:** Als Systemadministrator möchte ich **Prometheus zur Überwachung der Systemressourcen einrichten**, um Engpässe frühzeitig zu erkennen.
  + ✅ Akzeptanzkriterien: Prometheus sammelt CPU-, RAM- und Netzwerkmesswerte aller Pi’s.
  + 🛠 Tasks: Installation von Prometheus, Konfiguration der Datenquellen.
* **User Story 4.2:** Als Projektteam möchte ich **ein Dashboard in Grafana erstellen**, um Systemmetriken in Echtzeit zu visualisieren.
  + ✅ Akzeptanzkriterien: Grafana zeigt relevante Daten an.
  + 🛠 Tasks: Installation von Grafana, Anbindung an Prometheus, Dashboard-Erstellung.

# Epic 5: Backup-Strategie für die Infrastruktur

* **User Story 5.1:** Als Administrator möchte ich eine **automatisierte Backup-Lösung** für alle Pi’s einrichten, um Datenverlust zu vermeiden.
  + ✅ Akzeptanzkriterien: Backups werden automatisch und regelmäßig erstellt.
  + 🛠 Tasks: Installation von Rsync/Duplicity, Skripterstellung für automatische Backups.
* **User Story 5.2:** Als Administrator möchte ich eine **Backup-Wiederherstellung testen**, um sicherzustellen, dass die Backups verwendbar sind.
  + ✅ Akzeptanzkriterien: Erfolgreiches Wiederherstellen der Daten aus einem Backup.
  + 🛠 Tasks: Durchführung eines Test-Backups und einer Wiederherstellung, Dokumentation des Prozesses.

# Epic 6: Sicherheitsmaßnahmen & Optimierungen

* **User Story 6.1:** Als Administrator möchte ich den **SSH-Zugriff absichern**, um unbefugten Zugriff zu verhindern.
  + ✅ Akzeptanzkriterien: Nur autorisierte Benutzer können sich einloggen.
  + 🛠 Tasks: SSH-Keys einrichten, Zugriffsbeschränkungen konfigurieren.
* **User Story 6.2:** Als Administrator möchte ich **Firewall-Regeln einrichten**, um unerwünschte Zugriffe zu blockieren.
  + ✅ Akzeptanzkriterien: Nur notwendige Ports sind offen.
  + 🛠 Tasks: Firewall (UFW oder iptables) konfigurieren, Sicherheitsprüfung durchführen.
* **User Story 6.3:** Als Administrator möchte ich die **Backup-Daten verschlüsseln**, um deren Sicherheit zu gewährleisten.
  + ✅ Akzeptanzkriterien: Backups sind nur mit einem Schlüssel oder Passwort lesbar.
  + 🛠 Tasks: Implementierung von Verschlüsselung mit GPG oder LUKS.

# Epic 7: Automatisierung & Continuous Deployment

* **User Story 7.1:** Als Entwickler möchte ich eine **CI/CD-Pipeline einrichten**, um die Website automatisiert zu deployen.
  + ✅ Akzeptanzkriterien: Änderungen an der Website werden automatisch getestet und deployed.
  + 🛠 Tasks: Einrichtung einer CI/CD-Pipeline mit GitHub Actions oder GitLab CI.
* **User Story 7.2:** Als Administrator möchte ich **die Infrastruktur mit Ansible/Terraform automatisieren**, um manuelle Konfigurationsaufwände zu reduzieren.
  + ✅ Akzeptanzkriterien: Pi’s und Konfigurationen können automatisiert bereitgestellt werden.
  + 🛠 Tasks: Ansible-Playbooks oder Terraform-Skripte erstellen.