Stop Sharing, Start Scaling

On-Demand Managed Kubernetes Environments

Who we are

Lukas Hoehl

- Software Engineer at STACKIT
- Working on Managed Kubernetes Service
- Network enthusiast and Golang lover

@hown3d

lukas.hoehl@stackit.cloud

STAC

A Brand of Schwa

Who we are

Marcel Boehm

- Cloud Engineer at inovex
- Has been at SKE for 2 years
- Can be found in the mountains when not writing k8s operators

@maboehm

marcel.boehm@inovex.de

inovex

Replicate complex cloud based software stacks for development purposes

STACKIT Kubernetes Engine

❤️

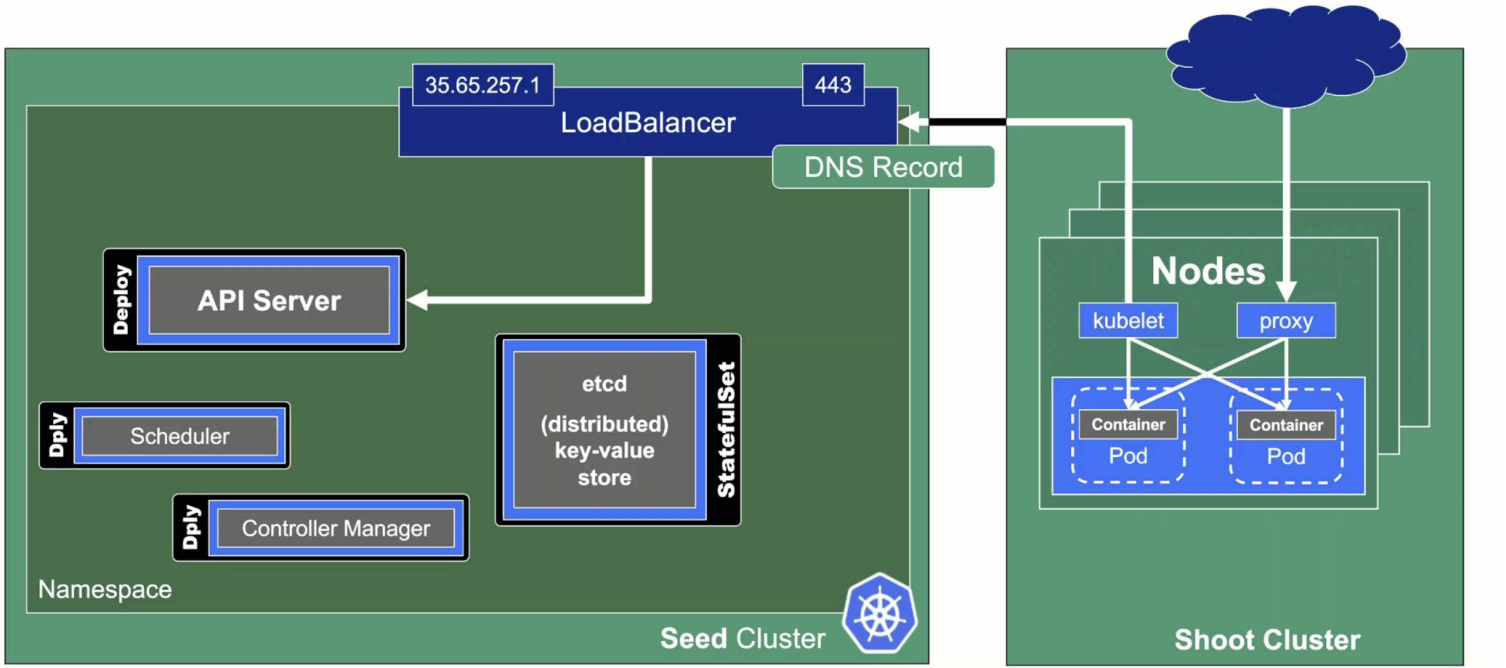
Gardener

STACKIT Kubernetes Engine ❤️ Gardener

- Active members of the community since 2019
- Part of the technical steering committee
- Build software to run Gardener on STACKIT

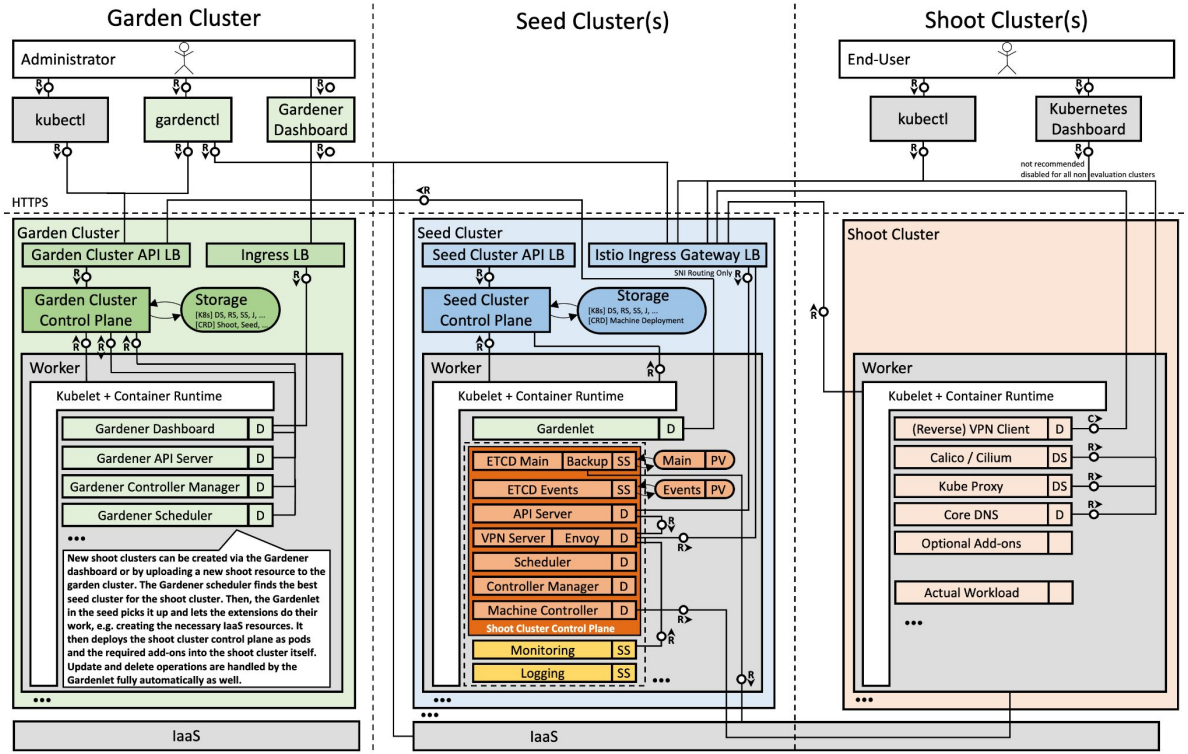
Gardener Hackathon 2024 Winter edition

Gardener runs Kubernetes clusters in Kubernetes



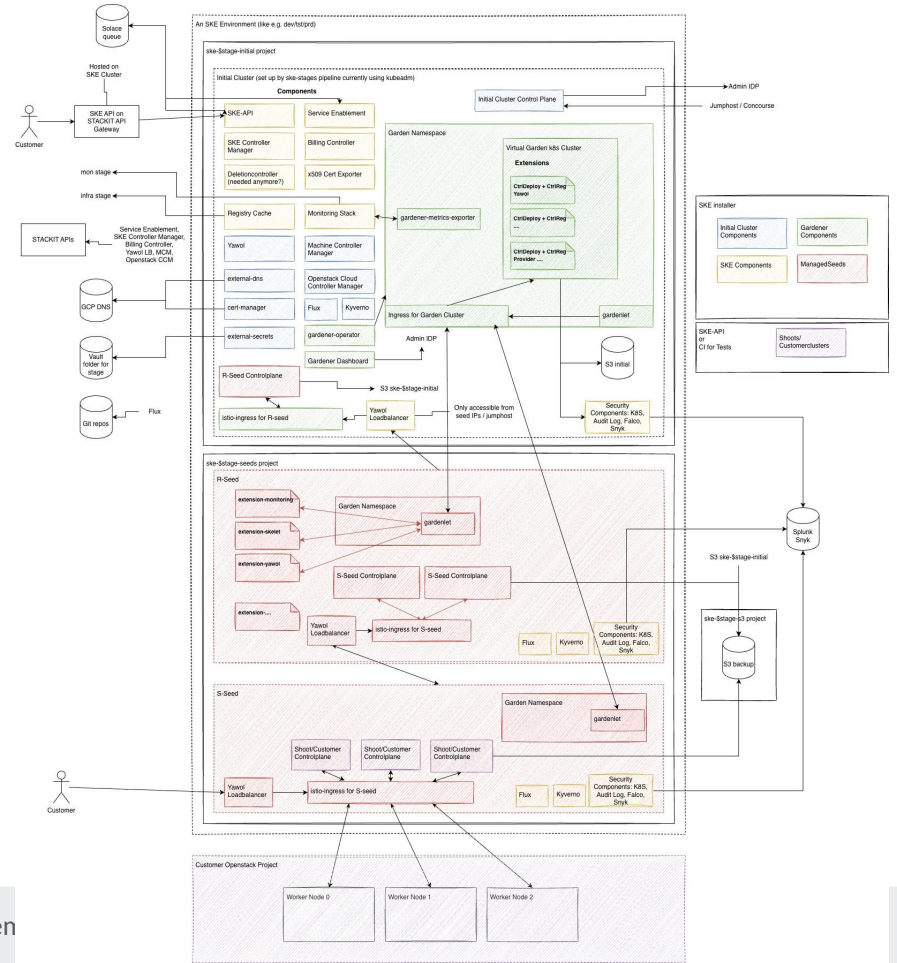
Gardener Architecture is complex

- Kubernetes as a service
- Leverages Kubernetes paradigms to deploy master components
- Manages worker nodes of clusters
- Create clusters in a declarative way



SKE Architecture is even more complex

- 15+ software components across 6 repositories that are developed standalone from Gardener
- Automatic cluster creation via API
- 100+ Flux Kustomizations and HelmReleases



Replicating our cloud setup on a local machine was hard

Why don’t you just run a subset of services locally?

- Interaction with Kubernetes principles like admission webhooks and reconcilers
- Gardener itself is already around 100+ pods in local setup
- Testing E2E

Prerequisites

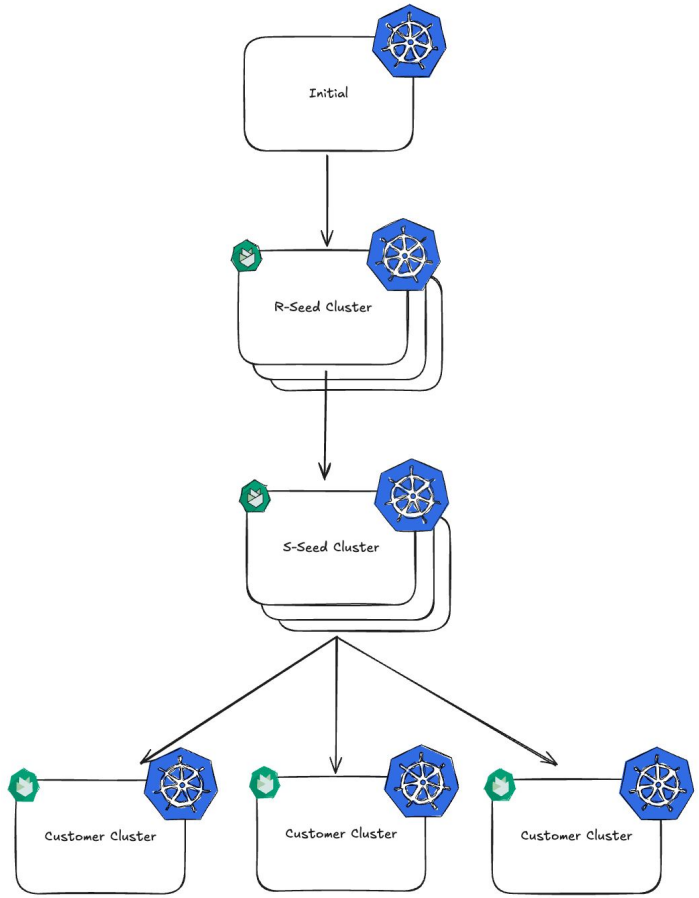
- Make sure that you have followed the Local Setup guide up until the Get the sources step.
- Make sure your Docker daemon is up-to-date, up and running and has enough resources (at least `8` CPUs and `8Gi` memory; see here how to configure the resources for Docker for Mac).

> Please note that 8 CPU / 8Gi memory might not be enough for more than two `Shoot` clusters, i.e., you might need to increase these values if you want to run additional `Shoot` s. If you plan on following the optional steps to create a second seed cluster, the required resources will be more - at least `10` CPUs and `18Gi` memory. Additionally, please configure at least `120Gi` of disk size for the Docker daemon. Tip: You can clean up unused data with `docker system df` and `docker system prune -a` .

Gardener local setup prerequisites

Chain of Clusters

- Scale-out concept
- Initial cluster managed by Terraform and kubeadm
- Rest of hosting clusters managed by Gardener



This is too complex for a local machine

Just use a dev environment!

Where we come from

- Broken dev environment a lot of times
- Gardener was setup with Bash scripts and templated YAML
- Setting up a temporal environment was not an easy thing to do
    - only done for testing really risky changes


AFTER ALL... WHY NOT?
WHY SHOULDN'T I USE IT?
imgflip.com

Where we are

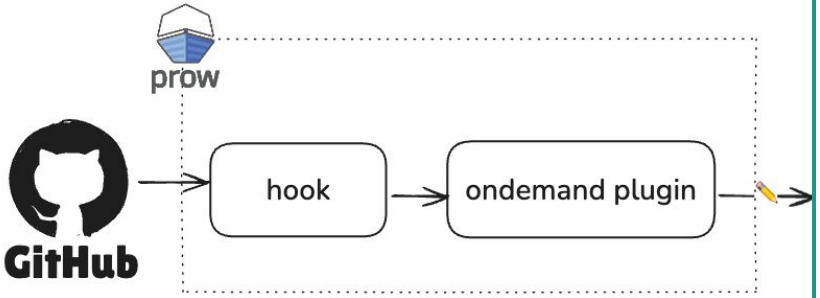
- Create Kubernetes cluster in Gardener to host Gardener (Gardenerception!)
  - Eat your own dog food
- GitOps principles with FluxCD
  - Automate setup of Gardener

Introducing On-Demand environments

- Replicate our prod environment as close as possible
- Allow to test changes in real world scenario
- Run the E2E test suite against this environment
- Break stuff without anybody noticing

Demo

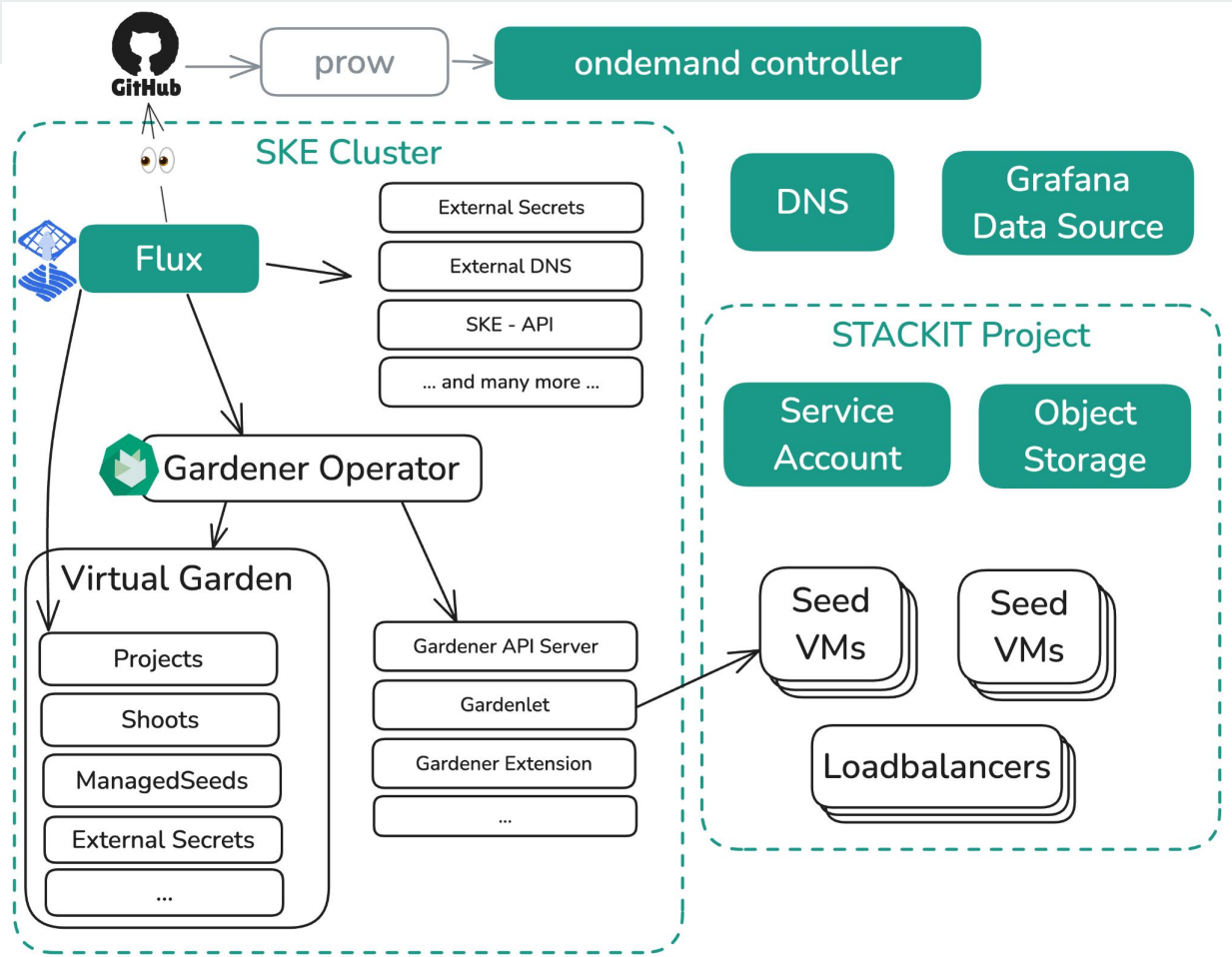
Request Flow




```yaml
apiVersion: ci.ske.stackit.cloud/v1alpha1
kind: OnDemandEnvironment
metadata:
  name: ond-tiger
spec:
  expiration:
    timestamp: "2025-11-27T15:13:57Z"
  requestedBy:
    github:
      issue: 1337
      kind: PullRequest
      organization: stackitcloud
      repository: ske-base
      user: maboehm
  sources:
    ske-base:
      ref:
        branch: add-dashboard-config
```

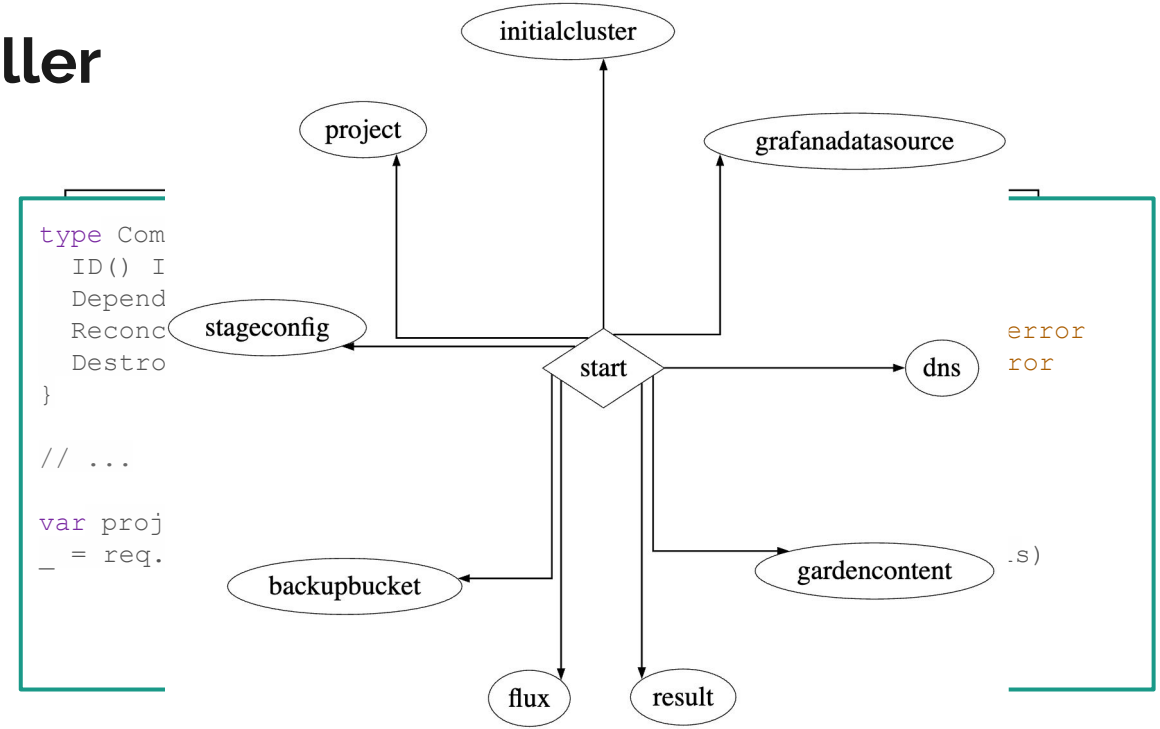
Deployed Resources

- On-Demand controller only deploys some Infrastructure Components
- The rest is created via GitOps



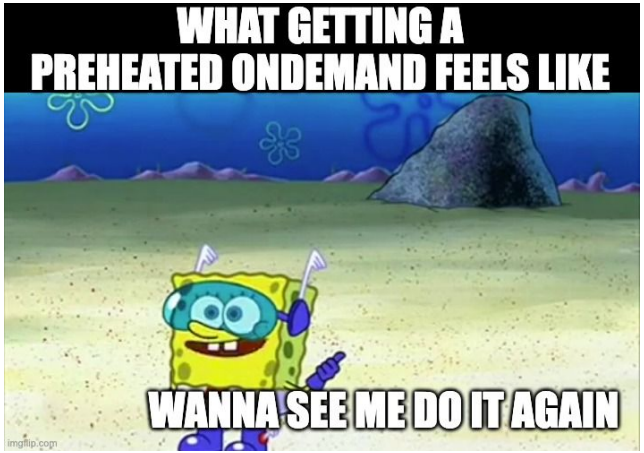
On-Demand Controller

- “Fan-Out” Controller
- Graph is executed in order
- “Component” declares its dependencies
- .. And can access data from it



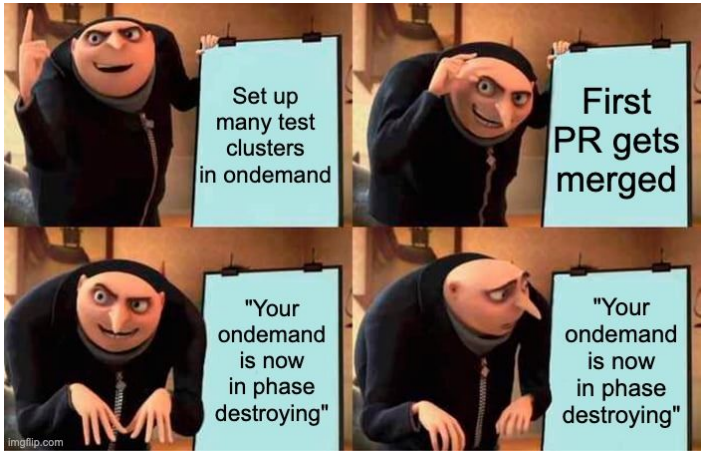
Other Features

Preheating



Time To Live

Auto-Deletion



Future Plans

GKE Cluster
Gardener Shoot
vCluster
AKS Cluster

with a PostgreSQL Flex instance

ArgoCD Application
Terraform Pipeline

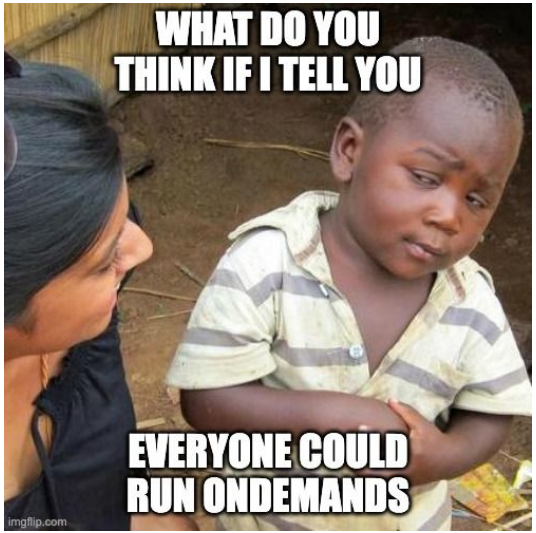
and an Azure Key Vault

Bitbucket
GitLab
STACKIT Git
Azure DevOps

“Create a **SKE Cluster in STACKIT** and point the **FluxCD Source** to my branch on **GitHub**”

Vision

“One day, every team can run On-Demands on their own”



Lessons Learned

- GitOps is key for simply creating another environment
- You have to treat your On-Demand as a “first class citizen”
- It should be deeply embedded in your workflows
- On-Demand environments should be like your DEV or QA Environment
- Encourage “short-lived” nature - it’s easier to create & delete than to maintain & update
- They are costly, but worth it

Stop Sharing your DEV,
Start Scaling, with On-Demands!