An Early Adopters Story About
SUSE Cloud Application Platform

Adfinis SyGroup - Switzerland
Nicolas Christener

CEO/CTO Adfinis SyGroup

nicolas.christener@adfinis-sygroup.ch

twitter.com/nikslor
linkedin.com/in/christener
Lucas Bickel

Developer @ Adfinis SyGroup
OSS dev by night

lucas.bickel@adfinis-sygroup.ch

twitter.com/hairmare
About Adfinis SyGroup

Since 2000

Berne, Basel, Zürich & Lausanne

Over 55 employees

Broad customer base

100% Open Source
Our Services

Engineering  Managed Services  DevOps  Development
Our Partners

- SUSE
- HashiCorp
- Microsoft
- splendid data
Switzerland
Stereotypical Swiss Icons
Also Swiss

OS4: Early Free Flying Quadcopter at a Swiss Federal Institute of Technology

- Collision avoidance
- Autonomous flight
- Flight planning
- Helped kickstart the current drone craze
Also Swiss

VS-Code: Source-code Editor by MSFT

- Most popular developer environment
- Built in Zürich
- By Erich Gamma and team
- Open source made in Switzerland
Not Swiss

Dedicated Regions by Public Cloud Providers

● Some countries have tailor-made solutions for their government customers - none in Switzerland so far

● Governance, compliance, and data protection regulators mandate a custom built solution for Swiss government customers

● We took the opportunity to build a solution in cooperation with SUSE
Let's talk about CAP baby, let's talk about you and me, ..., let's talk about CAP
From Docker to Cloud Foundry

● Docker introduced the masses to containers
● Container workloads require a container orchestration solution
● Kubernetes (K8s) is the de facto container orchestrator
● Cloud Foundry (CF) development started even earlier
● SUSE is bringing Cloud Foundry to the K8s ecosystem
Kubernetes 101

Master

Node

Container Engine

POD

Container

Node

Container Engine

POD

Container

Node

Container Engine

POD

Container

Node

Container Engine

POD

Container
CloudFoundry 101 (Eirini Style)

code → cf push → running app
Why is this so cool?
SUSE Cloud Application Platform (CAP) is the simplest Cloud Foundry distribution available
A Lightweight Cloud Foundry

Pivotal Cloud Foundry (PCF)
31 Nodes, 43 vCPUs, 122 GB RAM
http://pcfsizer.pivotal.io/#!/sizing/azure/2.2/small

Cloud Application Platform
11 Nodes, 22 vCPUs, 88 GB RAM
Small setup on Azure using AKS
A Portable PaaS Solution

SUSE Cloud Application Platform officially supports different K8s stacks

- SUSE CaaSP
- Amazon EKS
- Microsoft AKS
- Google GKE

This is not the cases for other PaaS solutions

- OpenShift is not supported on other K8s platforms
- Pivotal does not support plain K8s at the moment
A Developer Centric Solution

Cloud Foundry style

- Developers can use "cf push" and focus on code
- Sensible amount of settings
- Open Service Broker API was born in CF area
- Perfect for Spring-Boot devs

Kubernetes style

- Developers need to mess with "kubectl" and "s2i" and friends
- Large amount of settings
- Open Service Broker API still an alien
- Generic Infrastructure Solution
If one focuses on developer agility, Cloud Foundry (CF) is the answer!
A Story About a Somewhat Bigger Project
Let the Developers do Their Magic!

- Teams need a platform to deploy their tailor-made applications
- Time to market is a key factor in enabling customers success!
- DevOps mindset requires access to self-service capabilities
More Details About the Project

- Swiss organizations tend to run their own infrastructure
- "No one has access to my data" was the #1 strategy for many Swiss services - e.g. banking, pharmaceutical and other industries
- Swiss Government wants to keep the data in Switzerland as well
- Broad acceptance of public cloud offerings will not happen soon
Goals

● Provide a PaaS for Swiss federal offices

● Integrate provided services into existing service catalog

● Cloud-like billing, connected to existing SAP landscape

● We need to separate tenants physically in some cases

● Make developers and operators happy by building an awesome platform
Lovely Details Worth Mentioning

- Direct contact with SUSE product management allows influencing the future of the platform

- The shift to “Open Source first, upstream first” by SUSE was done at the right time for us. It enables us to help fix documentation for our customer at the proper upstream venues!

- We have the opportunity to help lay the foundation for the future of government cloud computing in Switzerland
Use Cases
Use Cases - Goals

- End-user self-service portal for non-dev customers
- Allow automating all the things
- Offer various in-cluster services
- Allow users to consume standard services w/o deploying to the cluster
- Everything needs to be billable
## Use Cases - User Scopes

<table>
<thead>
<tr>
<th>End User</th>
<th>“Internal” User</th>
<th>Operator</th>
<th>Platform Owner</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Can order services and pay for them</td>
<td>● May also be end user</td>
<td>● Maintains the cluster</td>
<td>● Owns government cloud strategy</td>
</tr>
<tr>
<td>● Does not have any escalated privileges</td>
<td>● Develops &amp; deploys apps using CF API</td>
<td>● Has full RBAC access</td>
<td>● Responsible for platform &amp; dev lifecycle mgmt</td>
</tr>
<tr>
<td>● Uses a well integrated self-service portal (HP-OO based)</td>
<td>● Automates everything from development to deployment &amp; day two operations</td>
<td></td>
<td>● Manages commercial &amp; financing aspects of the platform</td>
</tr>
<tr>
<td>● Time to market and elasticity are crucial</td>
<td>● May also use space on the development environment to learn about the platform</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use Cases - Solutions

- Service offerings exposed via Open Service Broker API
- High-order PaaS features from CAP exposed to users through CF API
- HP Operations Orchestration (HP-OO) integration as self-service portal
- Billing integration done by customer
  - In the future we plan on standardizing on cf-abacus if it reaches general availability
Lessons Learned
Integrating into an Existing Environment
Existing Architecture

**Compute**
- Hewlett Packard
- On premise

**Storage**
- NetApp NFS storage
- Cluster on NAS level

**Network**
- Edge: F5 Big-IP
- LAN: Cisco
Architecture Insights

- Deployment on bare-metal
- Reuse existing hardware
- Software defined ... where possible
Compute
Compute - Goals

- Compute is mostly a no-brainer however…

- Spectre/Meltdown/etc. led to the "separate physically" requirement
  - Customer with specific security demands get their own CaaSP / bare-metal setup

- We want to automate Velum (admin) node installs
  - CaaSP + CAP installation should be automated as much as possible

- Automation means faster time to market
Compute - Reality

- Installation of a CaaS cluster not fully automated out of the box
  - Velum node is usually installed manually
  - new Kubernetes nodes need to be assigned a system role

- Manual installation is not an option for a service provider

- With automation we can tick the "reproducible" checkbox
Compute - Solutions

- Complete CaaSP automation is achievable
  - Had to create some AutoYAST + Cloud-init configuration

- Integrated pipeline to set up the rest done by customer
  - Does other things like set up billing, backup, etc.

- We strive to make documentation & code available so others can use it
Networking
Networking - Goals

● Exposing an internal service should automatically configure the LB

● The F5 LB in front of the cluster is integrated into the stack

● Restrict network connections of Pods

● Pods should not be able to sniff traffic of their neighbours
Networking - Reality

● Flannel doesn't offer enough network restriction

● Network automation has its own pitfalls
  ○ processes, governance, licensing, etc.
  ○ F5 automation needs "F5 SDN services" (additional $$$)

● Workloads with specific security demands need their own CaaS / bare metal setup
Networking - Solutions

- Outgoing proxy was a challenge both for the deploy and for ops
  - was fixed upstream by SUSE
  - check “Using a Proxy Server with Authentication” in the deployment guide

- Waiting to switch CNI (from Flannel → Cilium)
Storage
Storage - Goals

- Customers want their own storage volume on the storage cluster
- Volume provisioning should be automated
- Existing NetApp infrastructure shall be integrated into the stack
Storage - Reality

- NetApp offers a K8s storage orchestrator (Trident)
  ○ Open Source & works like a charm - thanks NetApp!

- Current CaaS can't use Trident snapshot functionality
  ○ K8s volume snapshots are an alpha feature - would be nice to have
  ○ Snapshots are manageable out of band for now
Storage - Solutions

- If you have NetApp, go and try Trident
- Can't say much about other storage vendors for now
  - Any feedback from the audience?
- Can't say much about SUSE Enterprise Storage (SES) in this context
  - I'm sure Lars Marowsky-Brée et al. made SES + CAP = 💖
Backup Restore Strategy
Backup Restore Strategy - Goals

- Primary areas of focus:
  - Disaster Recovery (DR)
  - Business Continuity Management (BCM)

- Reinstall and restore state rather than restoring full “nodes”

- Stateful resources (storage, databases) are kept in dedicated external clusters
Backup

Restore Strategy - Reality

- Internal container images aren't part of the backup
  - Can be rebuilt using CI/CD
  - Nevertheless we keep backups of all images for faster restores

- We don’t have complete control over all images in a users container registry but their images need to be restorable

- Stateful services (i.e. MongoDB, Cassandra) in the cluster are a reality even though we would prefer to run those externally
Backup Restore Strategy - Solutions

- Multi-pronged approach
  - K8s State: Heptio Velero (formerly Ark)
  - CF State: cf-plugin-backup
  - Storage: PV snapshots through NetApp Trident
  - etcd State: DIY tooling as an additional fallback should Velero fail
  - Dedicated solutions for in-cluster services (i.e. MongoDB, Cassandra)

- All of these need to be understood and managed
Logging
Logging - Goals

- Centralized logging is a first-class citizen
- Batteries included for customers
- Existing customers can use tooling they are used to (i.e. Splunk)
Logging - Reality

- Splunk has really high operational expenses
- Log shippers for Splunk exist but aren’t integrated out of the box
- Hopping onboard an existing Splunk installation is harder than expected
Logging - Solutions

- Integrated tooling has support for shipping to Elasticsearch/Logstash/Kibana (ELK) so we use it
- K8s logs are shipped to ELK using fluentd
- CF logs get shipped to ELK using firehose nozzle
- If customer isn’t happy with ELK we can switch to Splunk at any time
Monitoring
Monitoring - Goals

- Establish key performance indicator (KPI) based monitoring
- Offer monitoring as a service for tenants
- No single point of failure in monitoring
- Expose metrics while adhering to policy of least privilege
- We want to use included best of breed tooling like Prometheus
Monitoring - Reality

- Prometheus needs to be monitored itself
- There are no easy ways to enforce authorization in Prometheus
- Prometheus and Grafana need quite some configuration to be useful
Monitoring - Solutions

● Use Grafana as frontend to Prometheus
  ○ Deployment of dashboards to Grafana is important to help users get up and running

● Deploy multiple Prometheus stacks for multi-tenancy
  ○ This leads to tolerable redundancy over various Prometheus databases

● An additional external Prometheus instance monitors the in-cluster Prometheus
Security
Security - Goals

- Best of breed security concepts and tooling are the only option
- Network isolation is a must
- We want to automate as much as possible
Security - Reality

- Network isolation isn't possible with Flannel
- Helm's tiller doesn't make it as easy as we would like on the K8s parts
- RBAC is a lot of work to set up
Security - Solutions

- NIST Special Publication 800-190 - Application Container Security Guide (let’s not invent our own standards here)

- Adapt existing cultural and technical conventions to a new highly dynamic operating model

- Infrastructure and customer apps are grouped by protection, needs help enforcing network isolation where needed, if needed we deploy an additional cluster for sensitive apps
Security - Solutions

- MicroOS as a small ephemeral base for nodes
- Vulnerability management by scanning containers in continuous integration (CI) process
- A tiller installation per “RBAC-Group” for K8s users
  - We’re looking forward to helm 3
- CF already has a quite robust security model
Enabling DevOps
Enabling DevOps - Goals

- DevOps as key to enabling automation
- Allows a high degree of self-service capabilities
- Both the CF API and K8s are available to users
Enabling DevOps - Solutions

- Paradigm shift when it comes to adopting existing processes into a highly dynamic container environment

- Continuous Integration and Deployment is mandatory

- Three classical environments to aid rollout and testing
  - development
  - integration
  - production
Enabling DevOps - Solutions

A defined reference architecture helps onboard new developers
Enabling DevOps - Solutions

● Users that want full control can use helm and co.

● Users that want an opinionated stack use CF tooling

● Trainings for the customer and users → https://ad-sy.ch/trainings
Outlook & Thanks
To the Cloud and Beyond

- Federate across multiple data centers
- Be the default PaaS provider for Swiss government customers
- The platform is available and builds the foundation for large digital transformation projects
- Looking forward to new cloud native services from ISVs
Thanks! Merci! Dankeschön!

Early adoption needed help from SUSE that we received in time

- Markus Wolf, Country Manager SUSE Switzerland
- Ralf Dannert, Systems Engineer SUSE
- Carsten Duch, Sales Engineer SUSE
- Jeff Hobbs, Director of Engineering CaaSP/CAP, SUSE
- plus countless other awesome chameleons