About making Choices – CaaS Platform 4 as SUSE's empowering of Kubernetes

Bettina Bassermann
Solution Sales Specialist DevOps
Bettina.Bassermann@suse.com

Rania Mohamed
Solution Architect, Services Consulting
Rania.Mohamed@suse.com

Who is SUSE?

- Founded in 1992
- Largest independent open source vendor as of March 2019
- Technology company
- Our Mission is to help customers to master the digital transformation through Open Source technology
- Innovating with Partners and communites
- Enterprise-Grade Support

Series about modern Application Development

- Software Development, Microservices & Container Management, a SUSE webinar series on modern Application Development
- Please find all SUSE Webinars here

https://www.suse.com/de-de/events/webinars

https://www.suse.com/c/author/rasadus/

Microservices – Is it the Holy Grain? A Perspective of a Developer

Container and Cloud Native Technologies – Why do we need them and what is so great about it?

Why Kubernetes? A Deep Dive in Options, Benefits and Usecasese

About making Choices – CaaSPv4 as SUSE's empowering of Kubernetes

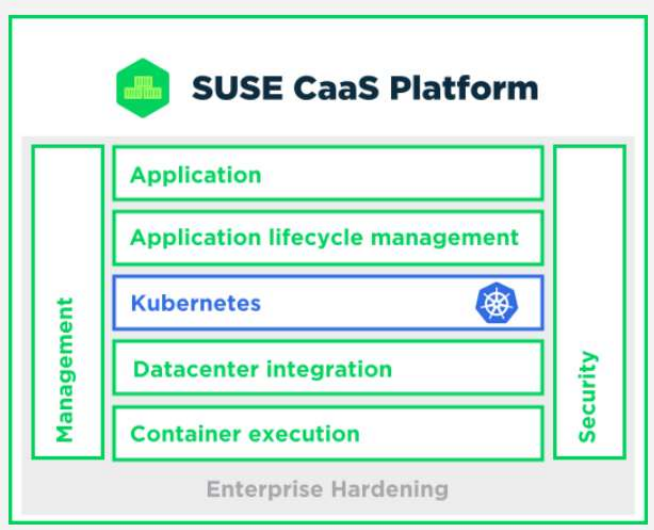
....stay tuned for the 2020 sessions with the Chamelion

Agenda

- SUSE CaaS Platform in a nutshell ☺
- SUSE CaaS Platform evolution
- SUSE Application Delivery suite
- SUSE CaaS Platform Architecture
- SUSE CaaS Platform Kubernetes Components
- Automation, proactive rather than just responsive ☺
- Deployment Models
- SUSE Application Delivery Console – Stratos
- Ecosystem DevOPs
- SUSE CaaS Platform delivered values
- SUSE CaaS Platform roadmap & the future ☺

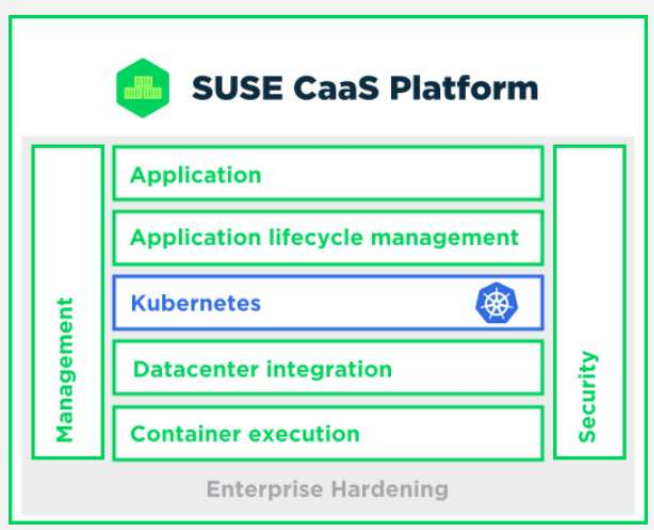
SUSE CaaS Platform – Container as a Service Platform – in a nutshell ☺

- CNCF Certified K8s distribution
- Supports the latest K8s version
- Targets both developers & operators
- Supports MSA/CNA and monolithic workloads
- Security is a top priority for both workloads and the platform ☺
- Target simplicity & Automation to the max ☺
- Follows & shares K8s big dream ☺



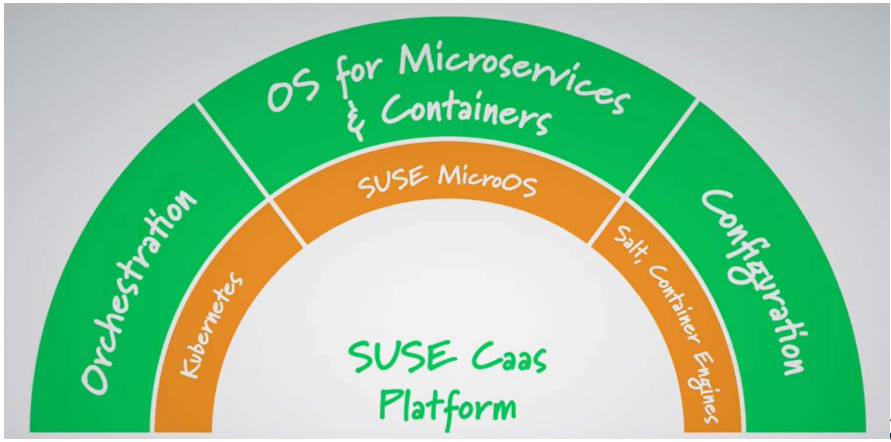
SUSE CaaS Platform evolution – Start of the story ☺

- Started back in 2016
- CaaS Platform had 4 major releases with dramatic changes in the targets and architecture
- Latest and greatest is named vNext and is known as SUSE CaaS Platform 4☺



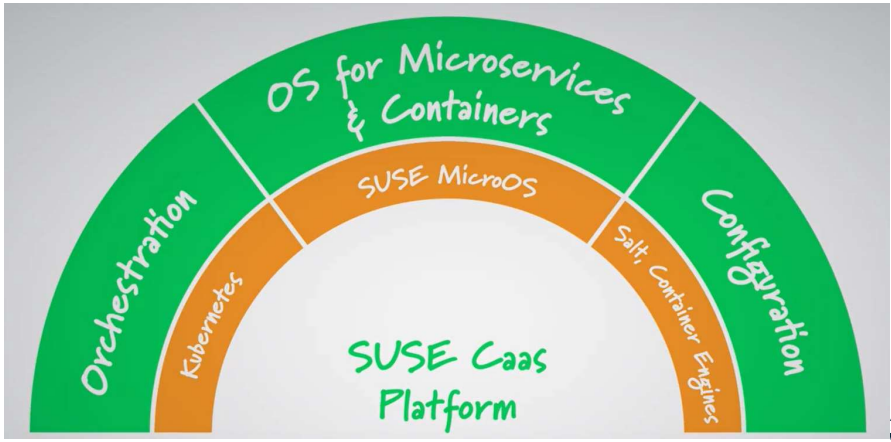
SUSE CaaS Platform evolution – Main releases – SUSE CaaS Platform1

- Formally released in 2017
- Based on MicroOS and not part of SUSE Linux OS
- Has its own Admin Dashboard
- Only uses salt automation
- Limitation in the supported HW/INF and storage



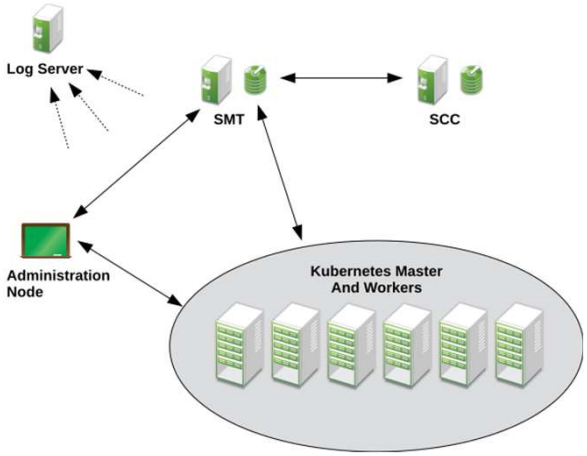
SUSE CaaS Platform evolution – Main releases – SUSE CaaS Platform2

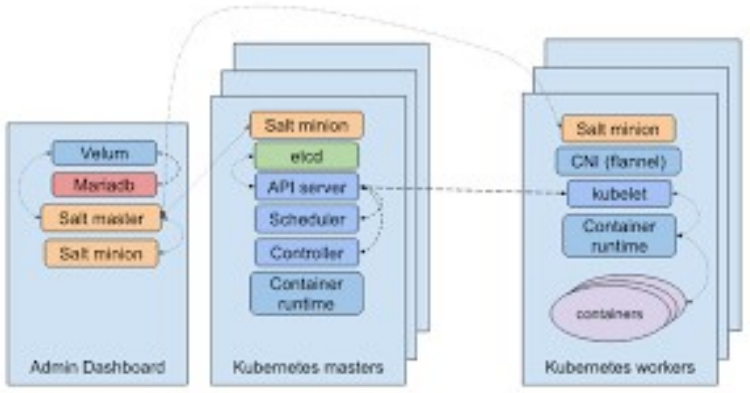
- Formally released in 2017-2018
- Based on MicroOS and not part of SUSE Linux OS
- Start support of a group of storages like SES and NFS
- Start support of a list of HW/INF such as KVM and IBM Z
- Still missing the complete story



SUSE CaaS Platform evolution – Main releases – SUSE CaaS Platform3

- Formally released in 2018
- Based on MicroOS and not part of SUSE Linux OS
- Solution is completely changed:
  - Admin node is added, having the embedded LDAP and Velum which is the dashboard
  - Salt is added to manage the patches to the nodes and changes
  - Bootstrapping is done using custom components
- More support for more and more INF/Storage as SOC, Vmware..
- Better security & control to the workloads and platform





Ok but why vNext? ☺

SUSE CaaS Platform evolution – SUSE CaaS Platform 3 challenges

- Complexity of patch management
- Misinterpretation of the MicroOS as a readonly OS
- Complexity of K8s version upgrades
- Complexity of the bootstrapping
- Complexity of the platform extendibility as it must be done using SALT
- Troubleshooting
- Running old version of K8s

Target of the platform is changing to follow the K8s big dream ☺

SUSE CaaSP evolution – CaaS Platform4

Thinking about the Future SUSE choose to take the next step towards hybrid cloud & multi-cloud and PaaS by going for the vNext CaaS Platform ☺

SUSE Application Delivery Suite

SUSE Application Delivery

- Group of solutions, such as Stratos, Metrics, SUSE Cloud Application Platform and SUSE CaaS Platform
- Facilitates app modernization and cloud app development with well defined migration strategies.
- Targets a growing development ecosystem
- Targets Ease of integration
- Supports modern development methodologies such as agile development
- Supports devops techniques and practices
- Helps with digital transformation journey

SUSE CaaS Platform 4

SUSE CaaS Platform 4 Architecture – Requirements

- Simplification to the max (solution, troubleshooting, manageability)
- Time of Market (must be fast in everything)
- Ease of K8s upgrade (no workload disruption and no downtime)
- Ease and simplicity of cluster setup, upgrade, migration and maintenance…
- Standardization is a priority (minimize custom components) – No vendor lock ☺
- Enable and secure multi-tenancy → Hybrid cloud ☺
- Better integration with other products
- Enables and supports devops
- Minimize the footprint
- Enable hybrid solutions and multi-cloud

Make it powerful and keep it simple ☺

SUSE CaaS Platform 4 Architecture – Principles

- **Automation – using SUSE AUTOYAST and Terraform**
- **Support public cloud**
- **Remove blockers and focus on the main dream, hybrid cloud**
  - Admin Node
  - Velum
  - MariaDB
  - Embedded LDAP
  - Embedded HAProxy
- **Follow K8s roadmap and future – use CRIO as the standard runtime engine**
- **Containerize the Control plane**

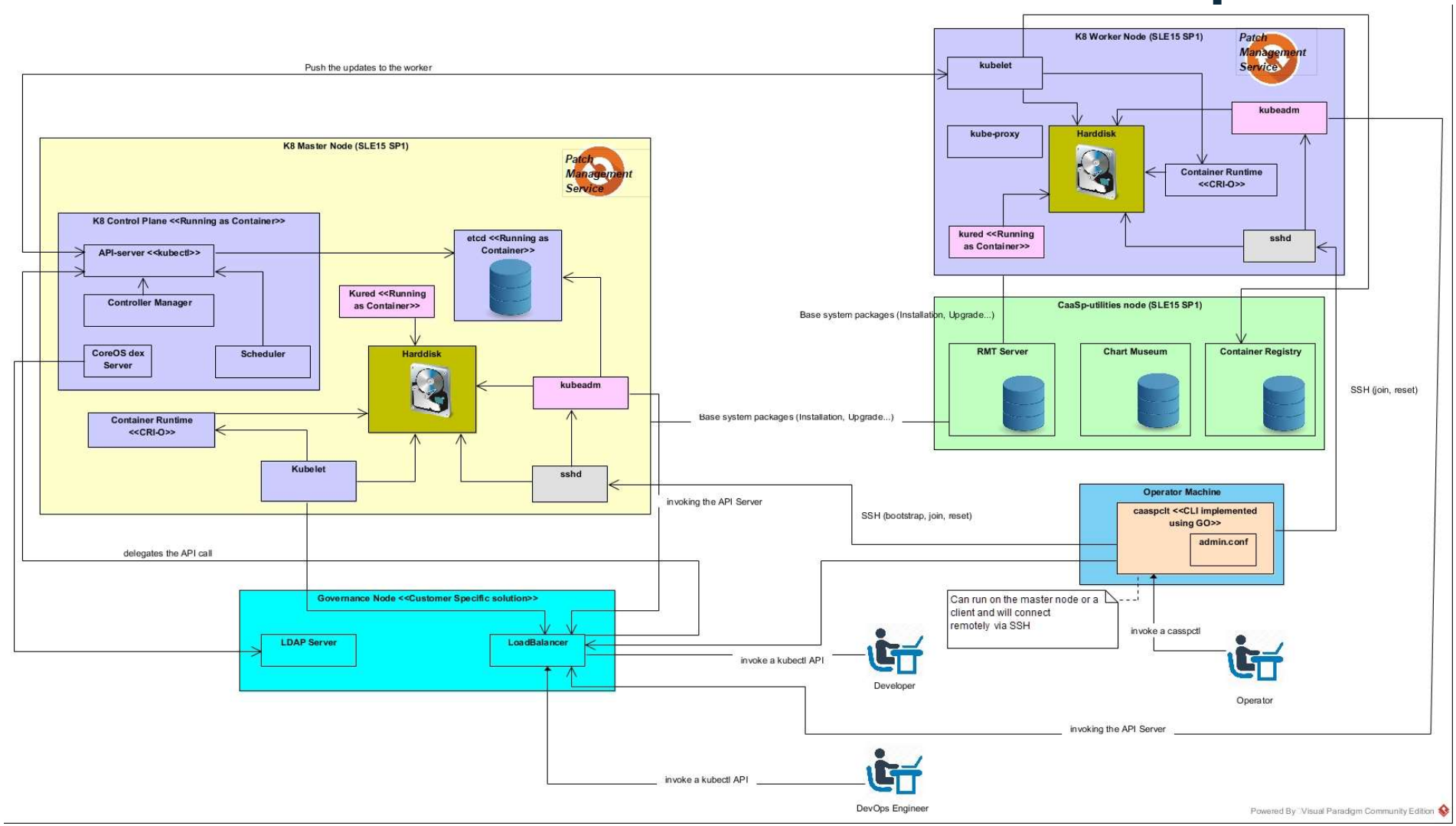
SUSE CaaS Platform 4 Architecture – Principles

- Use K8s standards tools for cluster setup and management, ex: kubeadm, kured
- Enable CaaSP on SLES 15
- K8s version upgrade must be managed separately and simply – no disruption at all to neither the platform nor the workloads
- Use application delivery console as a development and operator console – Stratos
- Use a ctl, to setup, operate and manage the cluster – Skuba
- Use cilium rather than Flannel to enable multi-tenancy and micro API-Gateways
- Enables ease of integration using central logging

SUSE CaaS Platform 4 Architecture – Principles

- **Architecture is API-centric**
- **NFR is a priority:**
  - Scalability
  - Maintainability
  - Speed and Automation
  - Extendibility
  - Standardization
- **Ease of backup/restore**
- **Ease of configurability for the cluster**

SUSE CaaS Platform 4 Architecture - blueprint



SUSE CaaS Platform 4 K8s Components

Master

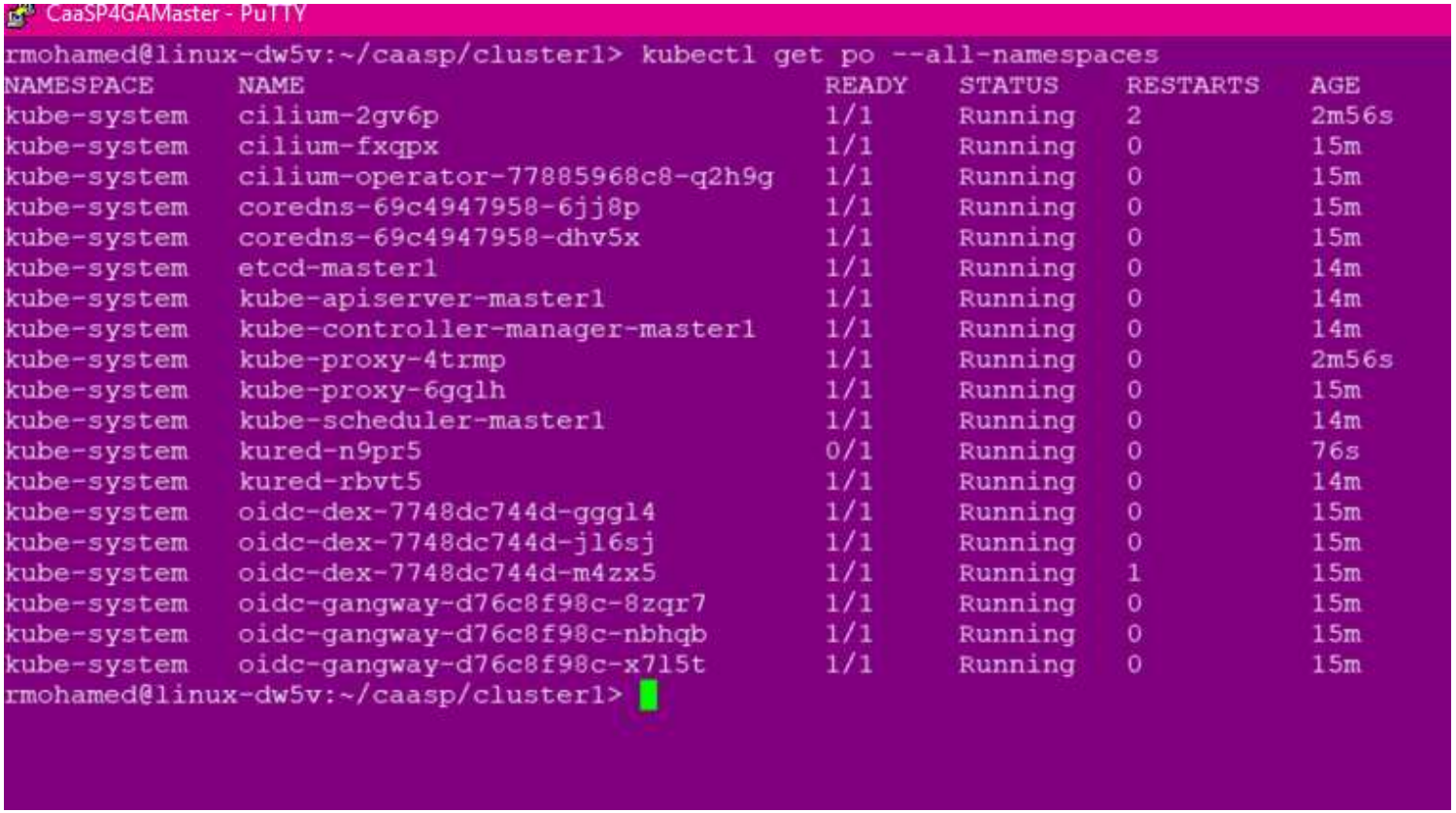
- Etcd → containerized
- Api server → containerized
- Main controllers → containerized
- Scheduler → containerized
- Kubelet → not containerized
- CRI-O → not containerized
- Dex & Gangway → containerized
- Kured → containerized
- Kubeadm →not containerized

Worker

- Kubelet → not containerized
- CRI-O → not containerized
- Kube-Proxy →containerized
- Kured → containerized
- Kubeadm →not containerized

Troubleshooting is much easier, it is part of K8s world ☺ customization is much easier; example admission controllers can be enabled or disabled using config map in K8s ☺

SUSE CaaS Platform 4 K8s Components



SUSE CaaS Platform 4 – Simplification & Automation

- Bootstrap a small cluster (1 master 3 workers) in less than 10 minutes using skuba ☺
- Skuba (https://github.com/SUSE/skuba) is a ctl enabling hybrid dream; manage multiple clusters in different inf from the same panel ☺

```
skuba cluster init --control-plane LB clusterName
skuba node bootstrap --user xxx --sudo --target nodeName nodeName -v10
skuba node join --role worker --user xx --sudo --target nodeName nodeName -v10
```

- SUSE CaaSP have multiple patterns to support different nodes types, example SUSE-CaaSP-Management
- Configurable Terraform to provision a cluster on an IaaS like VMWare and AWS

SUSE CaaS Platform 4 – Simplification & Automation – Terraform VMWare variables example

```
{
    "vsphere_datastore": "3PAR",
    "vsphere_datacenter": "PROVO",
    "vsphere_network": "VM Network",
    "vsphere_resource_pool": "CaaSP_RP",
    "template_name": "SLES15-SP1-GM-guestinfo",
    "firmware": "bios",
    "stack_name": "caasp-jenkins-v4",
    "masters": 1,
    "workers": 2,
    "username": "sles",
    "lb_repositories": {
        "sle_server_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Product-SLES/15-SP1/x86_64/product/",
        "basesystem_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Module-Basesystem/15-SP1/x86_64/product/",
        "ha_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Product-HA/15-SP1/x86_64/product/",
        "ha_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Product-HA/15-SP1/x86_64/update/",
        "sle_server_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Product-SLES/15-SP1/x86_64/update/",
        "basesystem_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Module-Basesystem/15-SP1/x86_64/update/"
    },
    "repositories": {
        "caasp_devel": "http://download.suse.de/ibs/Devel:/CaaSP:/4.0/SLE_15_SP1/",
        "suse_ca": "http://download.suse.de/ibs/SUSE:/CA/SLE_15_SP1/",
        "sle_server_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Product-SLES/15-SP1/x86_64/product/",
        "basesystem_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Module-Basesystem/15-SP1/x86_64/product/",
        "containers_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Module-Containers/15-SP1/x86_64/product/",
        "serverapps_pool": "http://download.suse.de/ibs/SUSE/Products/SLE-Module-Server-Applications/15-SP1/x86_64/product/",
        "sle_server_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Product-SLES/15-SP1/x86_64/update/",
        "basesystem_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Module-Basesystem/15-SP1/x86_64/update/",
        "containers_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Module-Containers/15-SP1/x86_64/update/",
        "serverapps_updates": "http://download.suse.de/ibs/SUSE/Updates/SLE-Module-Server-Applications/15-SP1/x86_64/update/"
    },
    "packages": [
    ],
    "authorized_keys": [],
    "ntp_servers": [
        "0.novell.pool.ntp.org",
        "1.novell.pool.ntp.org",
        "2.novell.pool.ntp.org",
        "3.novell.pool.ntp.org"
    ]
}
```

SUSE CaaS Platform 4 Deployment Models - Main

POC/Demo

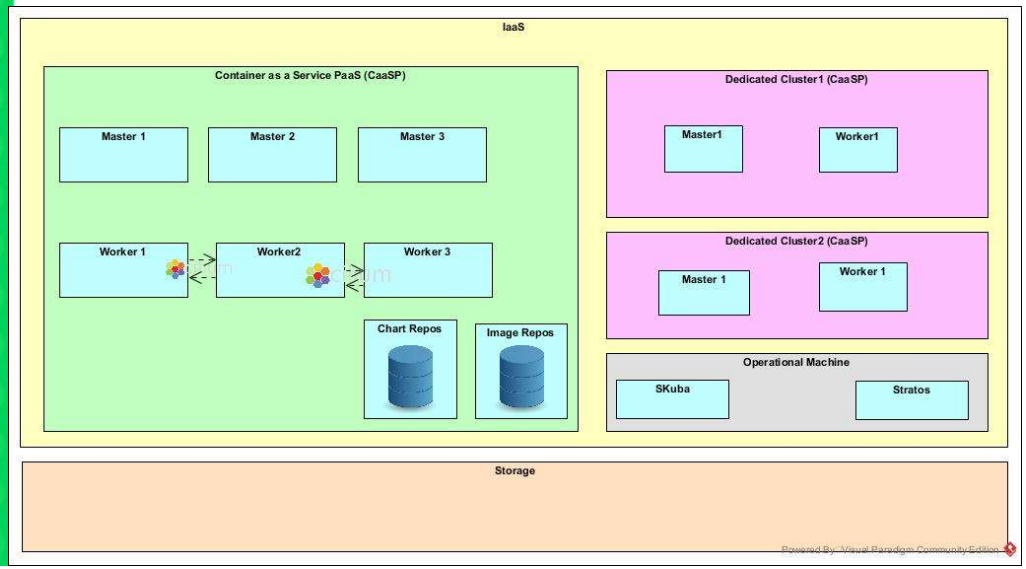
- Holds 1 master and 1 worker

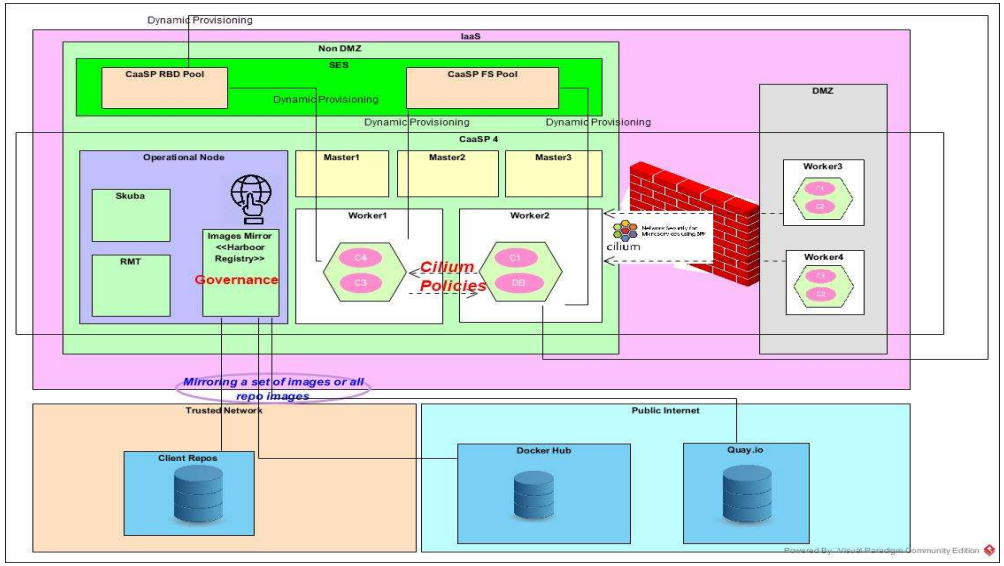
High Availability (HA)

- Minimum 6 nodes (3 master, 2 load balancers and at least one worker)
- More into HA of etcd, follow the quorum rule ☺

Air Gapped

- Isolated and controlled environment
- Requires RMT server and mirrored Images/charts Registries
- Recommended to setup it as HA





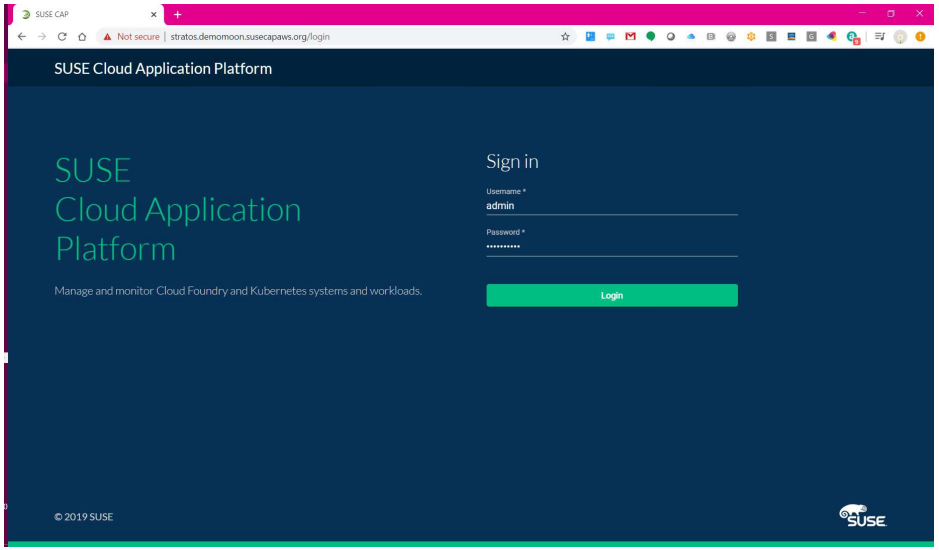
SUSE Application Delivery Console

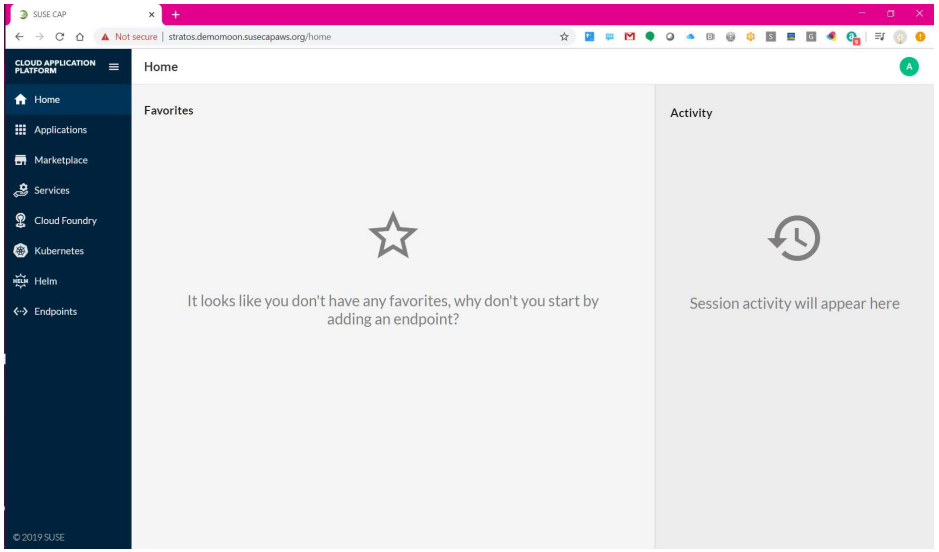
SUSE Application Delivery Console – Stratos

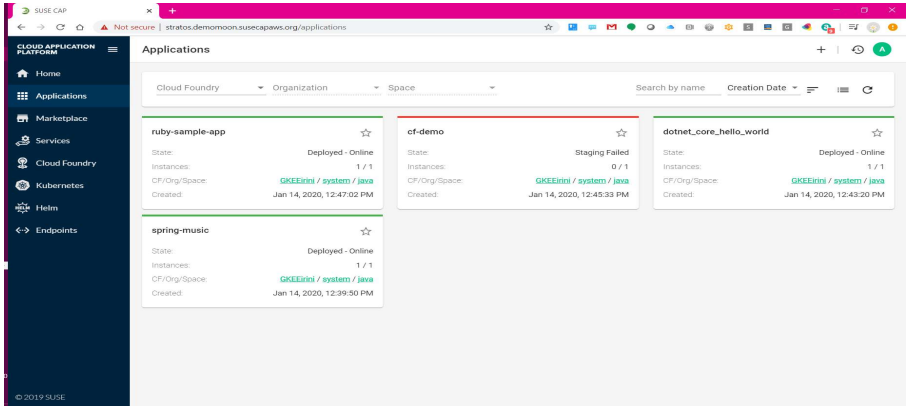
- Single Pane of Glass view ☺
- Target Developer to help developer push applications into K8s
- Help in troubleshooting and debugging the application
- Integrates with Prometheus & Grafana using our metrics component
- Enables backing services using helm charts
- Offer market place which may be integrated with public cloud services → hybrid & multi-cloud ☺
- Integrates with lots of tools/platforms

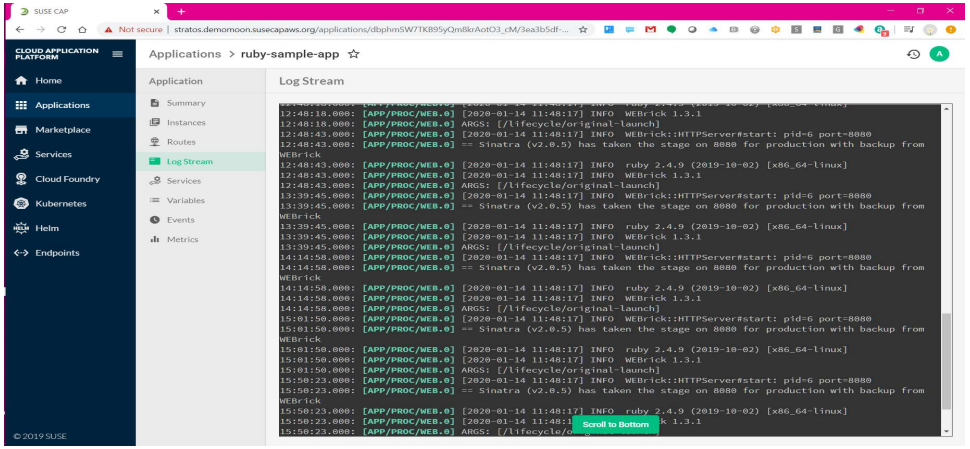
SUSE Application Delivery Console – Stratos Demo

SUSE Application Delivery Console – Stratos

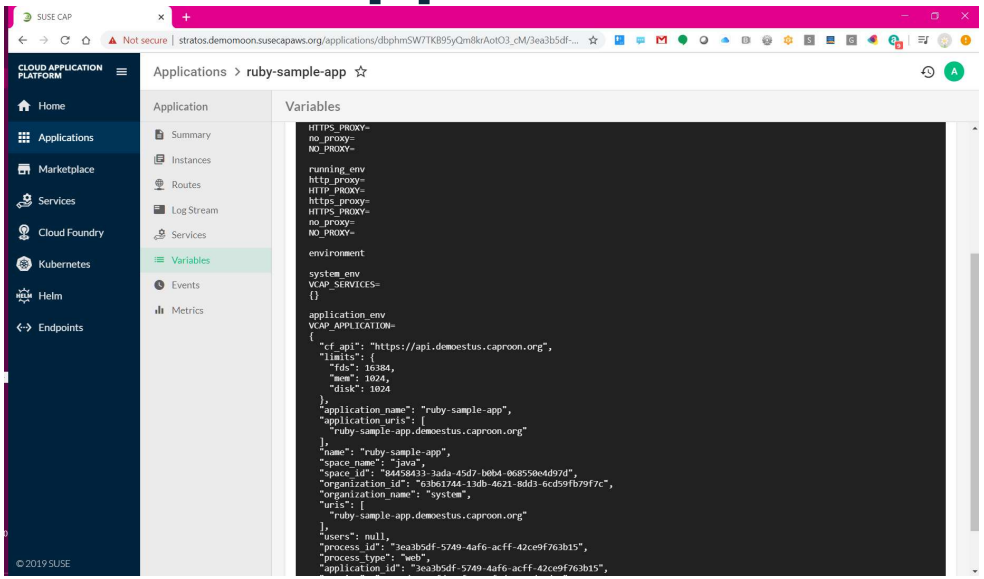


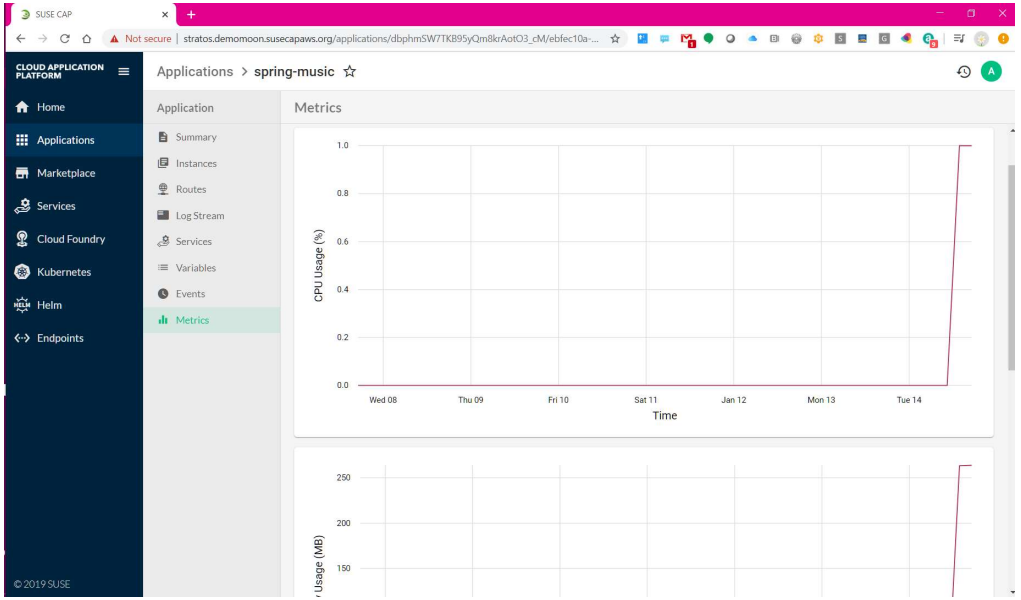


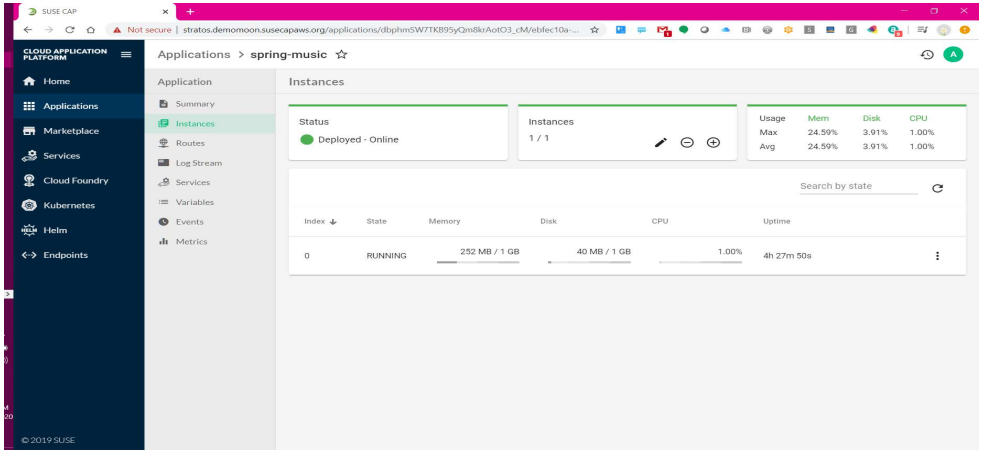


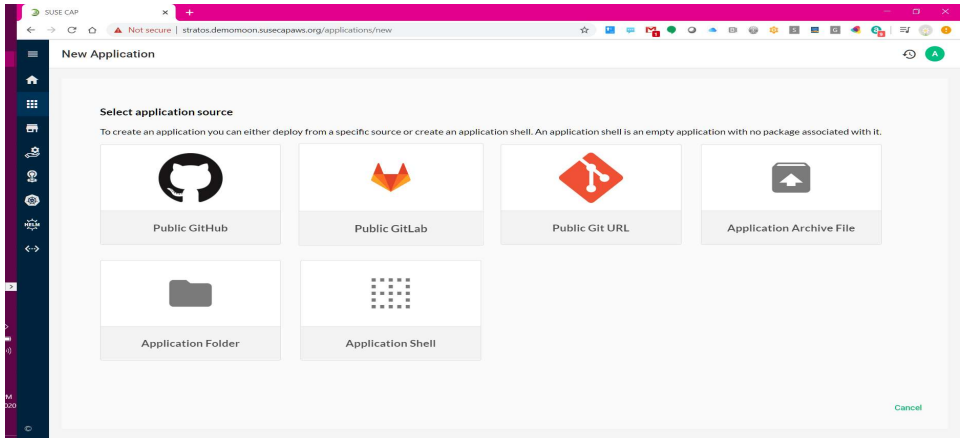


SUSE Application Delivery Console – Stratos

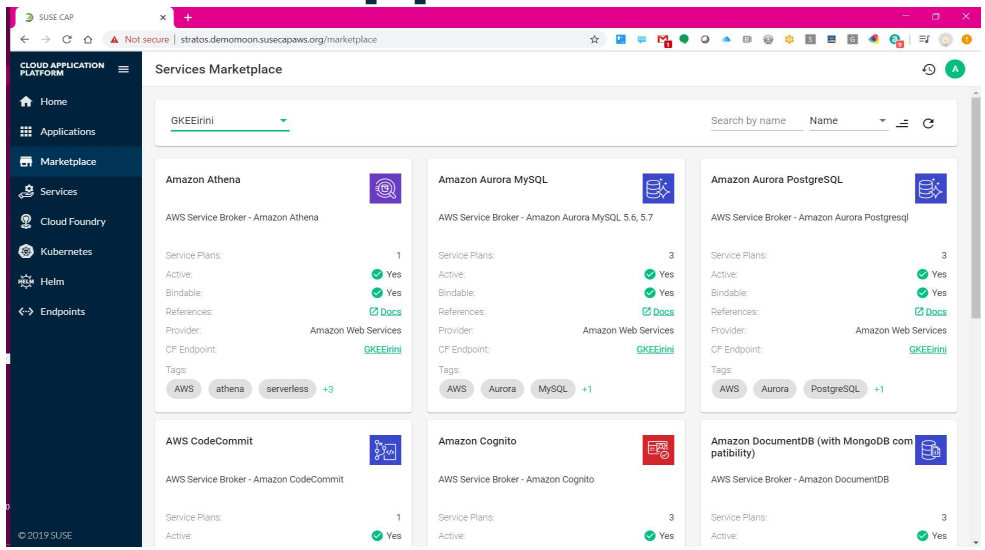


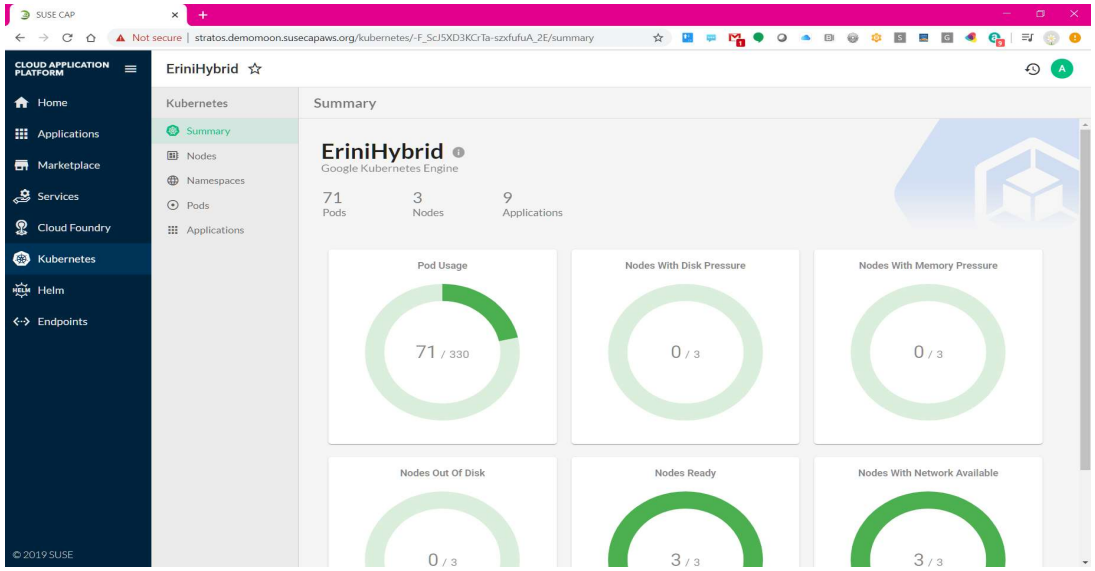


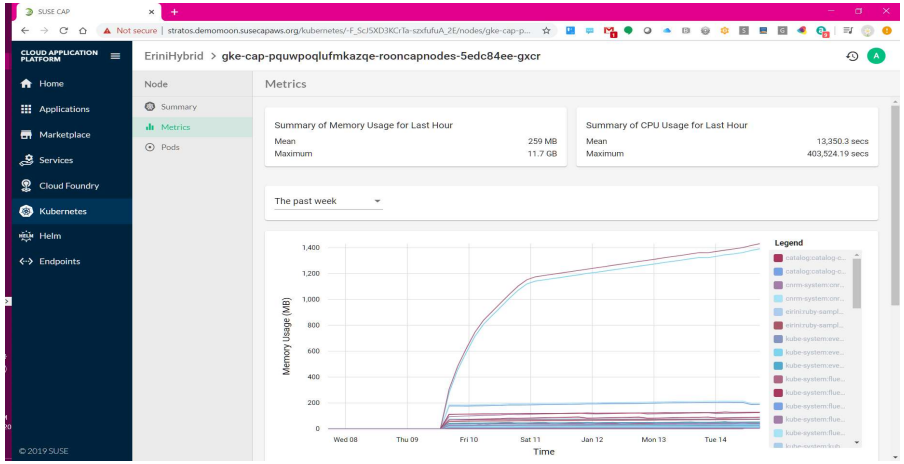




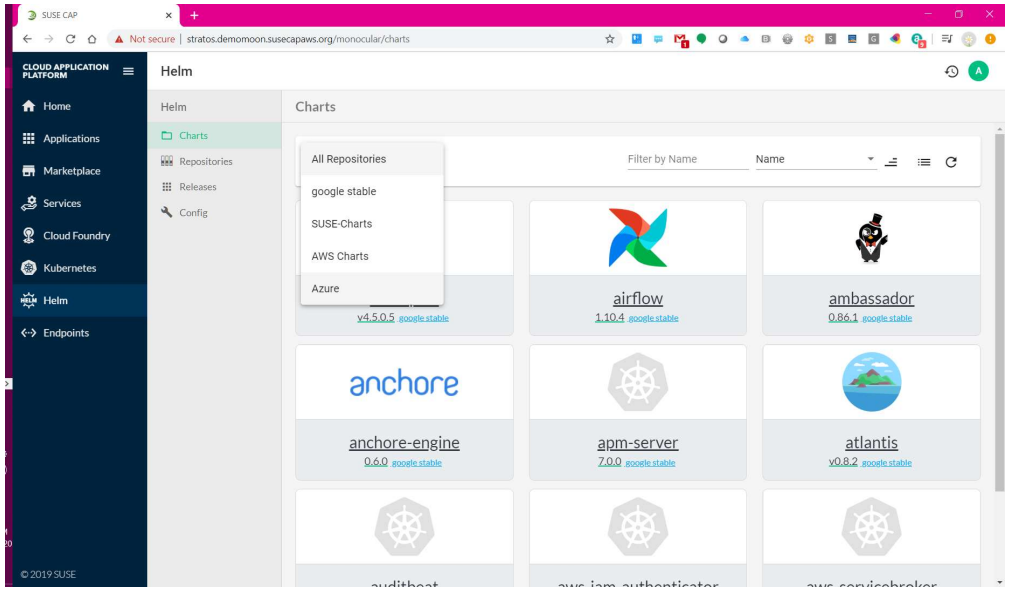
SUSE Application Delivery Console – Stratos

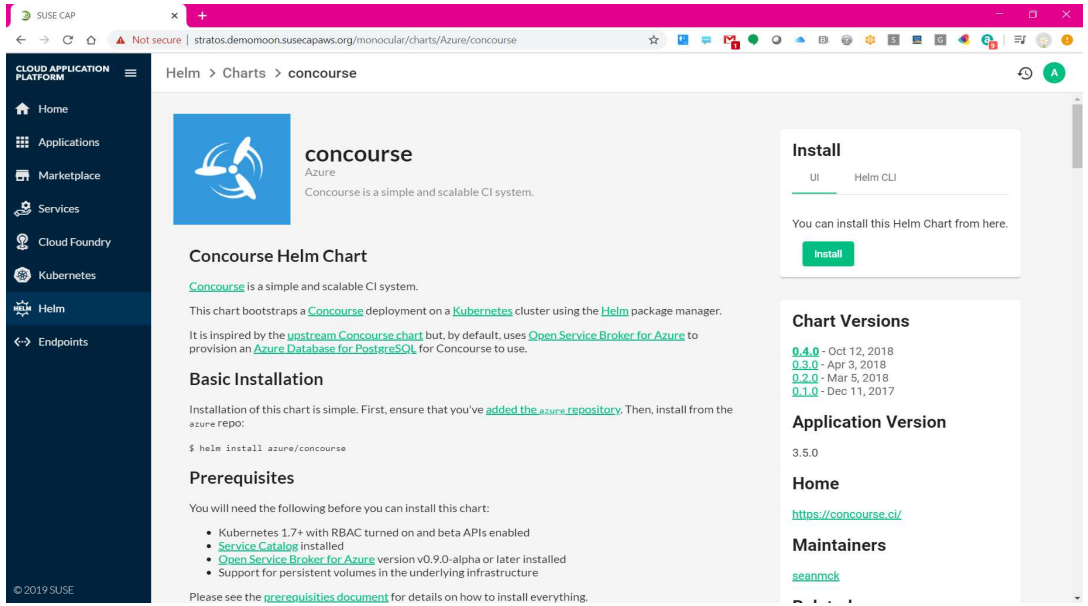


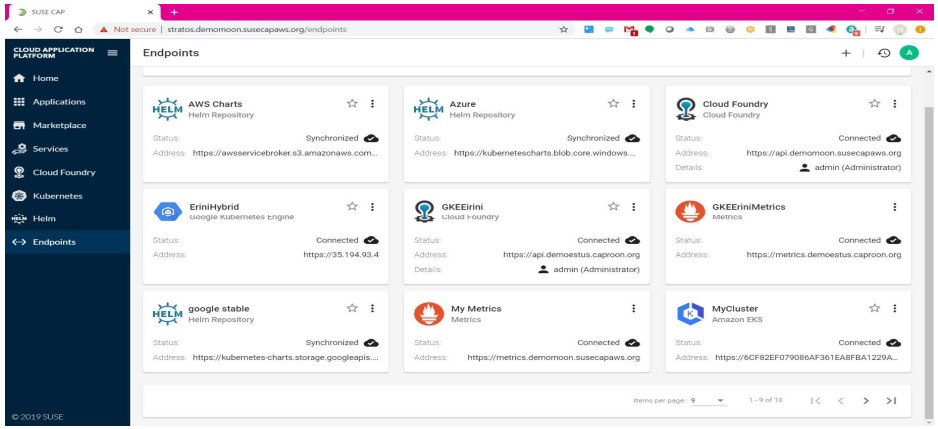


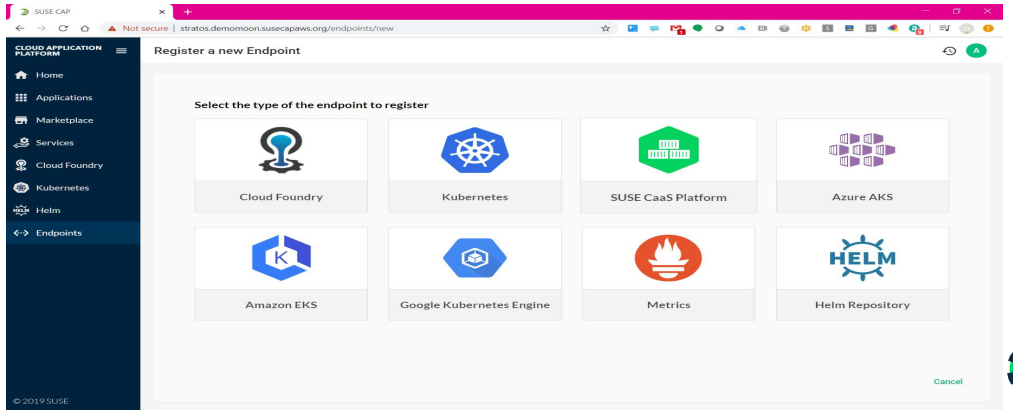


SUSE Application Delivery Console – Stratos









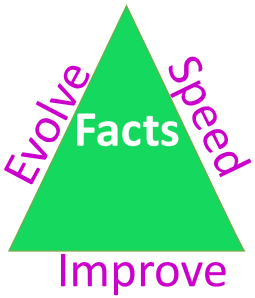
Ecosystem

Application Delivery Ecosystem – What & Why?

- **App delivery ecosystem is** group of tools/fw, processes and capabilities used by the team in order to deliver and operate services and applications delivery and its lifecycle → yes it is more of a Service Provider, helper, facilitator and a governor ☺
- **Not part of the platform as** it will differ from one client to another so it is better give choices ☺.
- Example of Ecosystem to SUSE CaaS Platform can be Infrastructure as a Service storage or services ☺.
- Characteristics of an Ecosystem is that it must be Self-service and programmable and continuously delivered → again and again our big dream ☺

You still have confusion let us go by categories and examples ☺

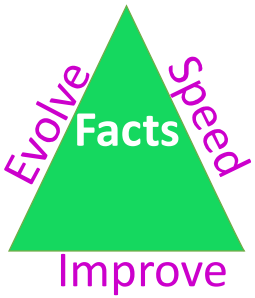
Application Delivery Ecosystem - DevOps



Set of Practices & culture philosophies aiming the high speed in development and ↑ in quality.

- **Agile Mgmt: JIRA, GitLab…**
- **Collaboration: Confluence…**
- **CD/CI tools: concourseci, JenkinsX..**
- **Configuration Management tools: GitLab, BitBucket..**
- **Automation testing FW: Selenium, Fitness, UFT…**
- **Security Testing: SonarQube, Vega..**
- **Performance Testing: LoadRunner, JMeter…**
- **Knowledge Management: Confluence…**

Application Delivery Ecosystem - DevOps



- IaC: CloudFormation, Terraform…
- Test & Defect Management: Jira…
- Repos/Registries: chartmuseum, Harbour…
- Build: gradle, maven, buildpacks, kaniko, buildah…

Application Delivery Ecosystem – Security & Analytics

Security

Set of tools to control and govern access, it may as well cover MSA compositions ☺.

- **API Gateway: apigee…**
- **IDM/SSO: MS ADFS, MS Active Directory …**

Analytics

Set of tools to gather facts, analyse it and enhance, and it is not just monitoring ☺.

- **Logging Aggregation: Fluentd, GrayLog, Elastic Stack, Splunk…**
- **Monitoring: Prometheus, DataDog…**
- **Visualization: Grafana…**
- **Analytics: Dynatrace, Elastic Stack**

Application Delivery Ecosystem – myths/confusion

- Can service mesh be used to embed and accomplish what is needed in ecosystem? Yes and even in the near future it is going to be more and more flexible and powerful
- Can a Plugin be used to implement requirements of the ecosystem or facilitate the integration? Yes, just like Cilium in case of API Gateway ☺
- Why doesn't the platform offer out of the box solution for the ecosystem requirement? It does but offers as well flexibility of the choice of integration which helps a lot in the digital transformation journey, like for example what SUSE CaaS Platform offer out of the box for logging and monitoring and other ecosystem capabilities ☺.

Values

SUSE CaaS Platform 4 – Delivered Business Values

- Time to Market

- Manageability

- Higher Customer Satisfaction

- Value Engineering

SUSE CaaS Platform 4 – Delivered Technical Values

- Agility
- No Vendor Lock
- Flexibility
- Simplicity
- Extendibility
- Integrability
- Integration
- Automation

SUSE CaaS Platform Future

SUSE CaaS Platform – Roadmap & Future ☺

4.0	4.1	4.2	4.3	5.0
New Architecture: Kubeadm based Command line first More flexible **Container Host OS** Codebase: SUSE Linux Enterprise 15 SP1 CRI-O as container engine **Orchestration** Kubernetes 1.15 Cilium network plugin	**Container Host OS** Codebase: SUSE Linux Enterprise 15 SP1 **New Features** Kubernetes 1.16 Prometheus and Grafana (metrics/visualization/alerts) Cilium 1.6 and Envoy (L7 policies) * Stratos UI (tech preview) AWS support (tech preview) *	**Container Host OS** Codebase: SUSE Linux Enterprise 15 SP1 **New Features** Kubernetes 1.17 Support on KVM Backup/restore and disaster recovery (using Velero) Support and marketplace availability on AWS, Azure, GCP Certificate view and refresh	**Container Host OS** Codebase: SUSE Linux Enterprise 15 SP1 **New Features** Kubernetes 1.18 GPU support (NVIDIA) Stratos UI (full support) CSI integration testing for SUSE Enterprise Storage vSphere storage CSI CSI self-certification for storage partners	**Container Host OS** Codebase: SUSE Linux Enterprise 15 SP2 **New Features** Kubernetes 1.19 Integrated VM orchestration (kubevirt) Istio service mesh Enterprise private registry NFS client provisioner

* - not in initial release

Overall themes

- **Continue to make Kubernetes easy to install, update, operate, and secure**
- **Multi-cluster, Multi-cloud**
- **Integration into customer environments (storage, networking)**

SUSE Cloud Application Platform – Roadmap

1.0 - 1.4

- Certified, Containerized Cloud Foundry Application Runtime (CFAR) on SUSE Linux Enterprise
- CFAR updates each point release
- Stratos UI web console for CFAR and Kubernetes
- Stratos Metrics: Prometheus database for platform metrics monitoring
- Supported on Azure AKS, Amazon EKS, and Google GKE
- Volume Service (NFS)
- OSBAPI Broker for Helm: MariaDB, PostgreSQL, Redis, and MongoDB
- App-AutoScaler
- CredHub: credential management
- Eirini: Kubernetes-native container scheduling (tech preview)

1.5

- Eirini updates: Closer feature parity with Diego scheduling (tech preview)
- Stratos: New Helm UI for browsing repositories and deploying charts
- Usage metering via Cloud Foundry Prometheus Exporter
- Deployment automation Terraform scripts for CSPs (AKS, EKS, GKE)

2.0

- All features and updates in 1.5 plus...
- **CF Quarks**: Kubernetes Operator for CFAR - better lifecycle management for upgrades and day two operations
- **Stratos:** Kubernetes features, Helm 3 updates,
- CSP marketplace availability

2.1

- **Services (tech preview):** SUSE- and partner-supported Helm charts exposing Marketplace services via OSBAPI broker
- **CI/CD (tech preview)**: Integrated Kubernetes-native CI/CD system (e.g. Drone, Argo, JenkinsX, etc.)
- **Minibroker:** Updates for Helm 3
- **Usage Metering**: Dashboards for usage tracking by CF Org and Space
- **Stratos:** additional Kubernetes and CF v3 API features. Helm 3 updates

2.2

- CI/CD: Further Stratos integration
- Log Aggregation: Elastic Search integration for CF and K8s logs
- Additional marketplace services

2.3

- **Serverless:** Function-as-a-Service in Cloud Foundry and Kubernetes, potentially using Knative (with existing FaaS framework or implemented within CFAR.

2.x

- Service Mesh / Istio & Envoy in CFAR (possibly scheduled sooner)
- Git interface: Gitea interface n Stratos
- Online IDE (VS Code via code-server)
- OCI image registry interface
- CF App SSO via Oauth2/UAA: Route service for limiting access to CF-hosted applications.
- Improved certificate management
- Windows-native .NET
- IBM zSystem support

Q&A

Thank you

Unpublished Work of SUSE LLC. All Rights Reserved.

This work is an unpublished work and contains confidential, proprietary and trade secret information of SUSE LLC. Access to this work is restricted to SUSE employees who have a need to know to perform tasks within the scope of their assignments. No part of this work may be practiced, performed, copied, distributed, revised, modified, translated, abridged, condensed, expanded, collected, or adapted without the prior written consent of SUSE.
Any use or exploitation of this work without authorization could subject the perpetrator to criminal and civil liability.

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. SUSE makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. The development, release, and timing of features or functionality described for SUSE products remains at the sole discretion of SUSE. Further, SUSE reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All SUSE marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.