1. Introduction

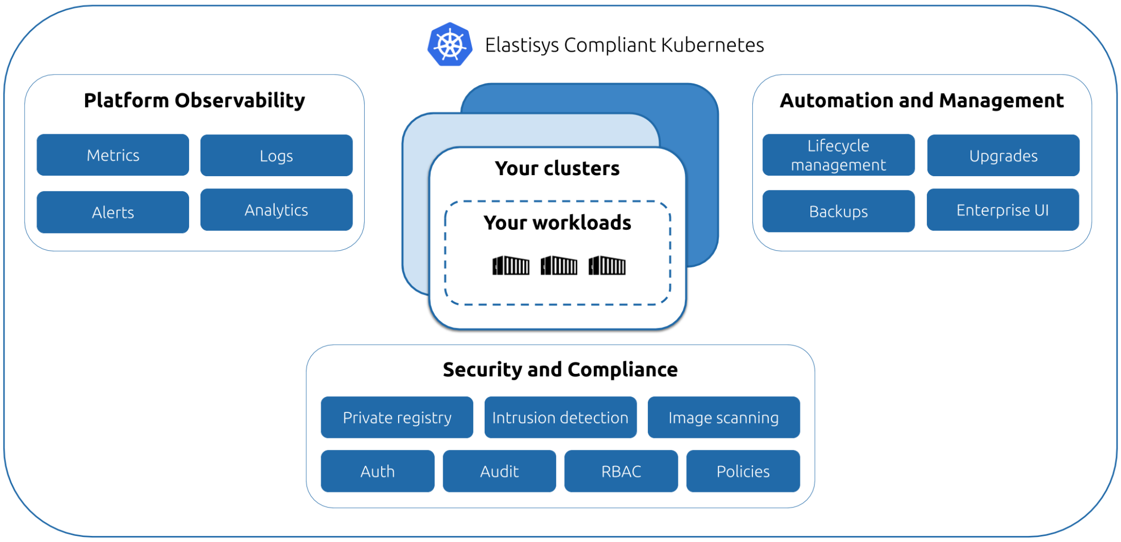
Cloud native technologies such as containerization, microservices, and immutable infrastructure empowers organizations to build better software, faster. In particular, Kubernetes allows more rapid deployment of new features to production as well as more efficient IT operations. In sectors where security and/or regulatory compliance are critical, adoption of cloud native technologies is a challenging task, given that technical implementation of common regulatory frameworks predate cloud native technologies.

Compliant Kubernetes (CK8s) is a proven, stable and secure Kubernetes platform built on open source cloud-native components. In addition to what is included in a "vanilla" managed Kubernetes Service, Compliant Kubernetes brings the following value:

- Worry-free container operations with platform managed 24/7 in ISO certified, European data centers.
- Pre-configured best practice security tooling to reduce compliance burden for frameworks such as ISO-27001, GDPR and PCI-DSS.
- Makes it easier to stay secure and compliant over time by enforcing policies across the whole software development lifecycle without restricting developers.
- Lessens the audit burden by providing detailed and easy to access audit trails.
- Makes applications easier to manage from an operations, compliance and security perspective by providing an enterprise UI that acts as a single point of entry to all relevant tools, policies, and configuration.
- Decreases the operational burden by managing all additional components required for a secure and compliant Kubernetes environment such as observability (logging, monitoring, auditing), authentication, secret management, intrusion detection, vulnerability scanning and a private container registry.

The following section describes the features of Compliant Kubernetes and outlines the architecture:

Features



The Compliant Kubernetes platform includes the following features and capabilities:

Security and compliance

- Private container registry
- Intrusion detection systems (IDS) for alerting in case of breaches
- Automated image vulnerability scanning
- Integration with authentication providers such as Active Directory, SAML, and OIDC, e.g. Google authentication
- Audit logging in the Kubernetes API server to track activities in the Cluster
- Role based access control (RBAC)
- Compliance policy enforcement
- Secret management

- Automated certificate handling
- Network segregation (network zones and isolation east-west traffic)
- Network isolation and restrictive firewalls, allowing only permitted network traffic into the platform. Inbound traffic to the Cluster is securely handled using the Nginx ingress controller

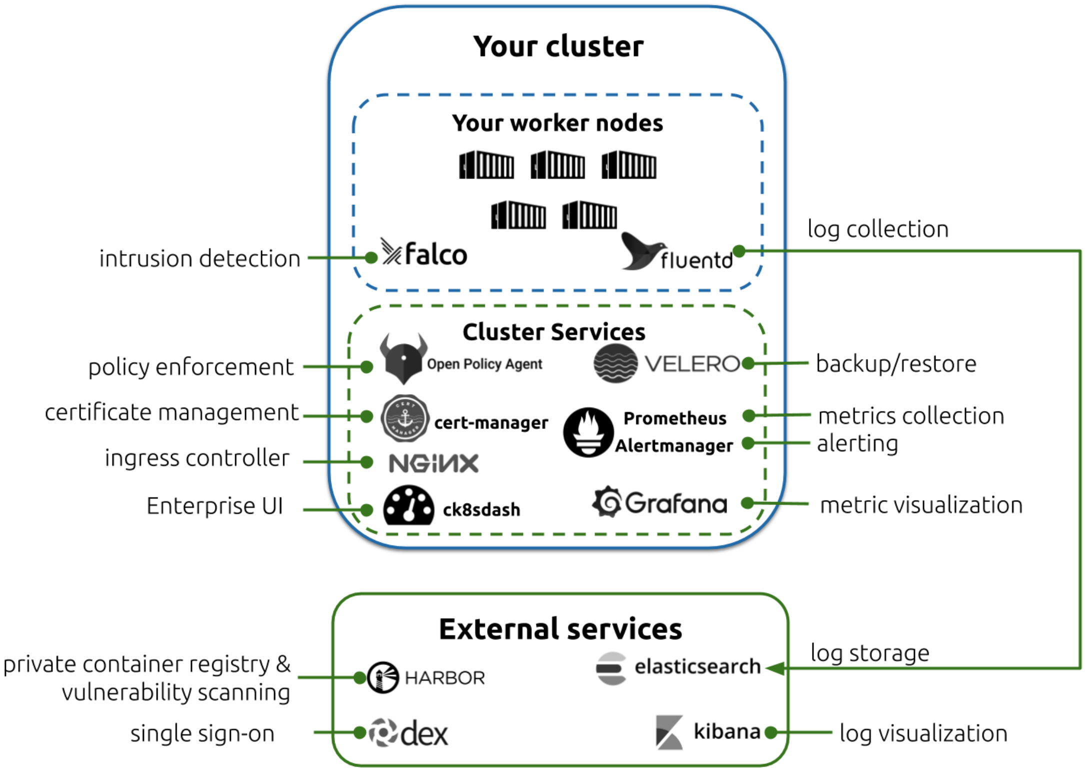
Platform observability

- Monitoring of Compliant Kubernetes platform resource usage
- Alerting based on monitoring data
- Log aggregation
- Analytics based on collected logs

Automation and management

- Continuous updates/patches of the Kubernetes platform
- Continuous updates/patches of Cluster Services and External Services
- Backups and disaster recovery
- Enterprise UI to control the Cluster and integrate with other services

2. Architecture



Compliant Kubernetes is delivered as a security-hardened Kubernetes Cluster. In addition to the standard Kubernetes control plane, Compliant Kubernetes includes a set of Cluster Services that run inside the Cluster as well as a set of External Services that run outside the Cluster. The above image illustrates the managed Kubernetes Service (with the Cluster Services and External Services shown in green color) and the Cluster where the customer can run applications (blue color).

The individual open source projects used to deliver the capabilities like log forwarding, backups, certificate management etc could be subject to change over time depending on how the ecosystem evolves and what is currently considered best practice. The latest infrastructure documentation and change log is available on request.

3. Capabilities and components

This section describes the setup of Compliant Kubernetes and its main capabilities.

The cluster

The compliant Kubernetes Cluster constitutes the following:

- The customer's worker Nodes
- 3 master Nodes for HA control plane
- Additional worker Nodes for Cluster Services (see below for list)

Worker nodes

The customer is free to choose the number and size of the customer's worker Nodes, up to 20 Nodes. Larger Clusters are available upon request. The number and size of the extra worker Nodes used for Cluster Services will depend on the customer's worker Nodes and applications.

Cluster Services

In addition to standard Kubernetes Services like CoreDNS and a network plugin that enforces Network Policies, the following Cluster Services are included in Compliant Kubernetes:

- Ingress controller (Nginx)
- Policy enforcement (OPA)
- Log forwarder (Fluentd)
- Metrics collection (Prometheus)
- Alerting system (Alertmanager)
- Intrusion detection (Falco)
- Certificate management (Cert-manager)
- Enterprise UI for Compliant Kubernetes (ck8sdash)
- Backup/restore Service (Velero)

External Services

In addition to the Services running in the Cluster, Compliant Kubernetes provides the following:

- Container registry, including vulnerability scanning (Harbor)
- Centralized logging Service (Elasticsearch/Kibana)
- Single sign-on Service (Dex)

These Services are delivered to the customer through a dedicated Kubernetes Cluster, consisting of:

- 3 master Nodes (HA)
- Worker Nodes

More details about the capabilities of all Services and their configuration are given below.
This setup represents the reference architecture, is meant for informational purposes, and may change over time.

Enterprise UI

Compliant Kubernetes comes with an enterprise UI to simplify the use of the Cluster and the Services. In the UI, customers can see the status of the Cluster including the Kubernetes resources (Pods, deployments, configmaps, etc.) that have deployed. Customers can also do normal Kubernetes operations such as adding resources, deleting resources, and scaling deployments. Each user logs in to the dashboard using our single sign-on Service and has different privileges in the UI based on their privileges in Kubernetes. Furthermore, the UI has integrations with some of the other Services, such as Grafana, Kibana, and Harbor.

Networking

Networking between the Kubernetes Nodes is performed over a *private network* to keep communication secure. Communication inside Kubernetes can be further controlled using network policies, which are enforced by a network plugin for Kubernetes. Exposure to the Internet is handled through an ingress controller (Nginx). Restricting and limiting access to and from the Internet is enforced through firewall rules and security groups in the datacenter. CoreDNS is used as DNS Service in the Kubernetes Cluster. Customers have access to the Kubernetes API over the Internet.

Certificate management

Automatic certificate management by cert-manager is available in the Cluster. Features include automatic certificate issuance and rotation from a certificate issuer such as Let's Encrypt CA for production grade certificates. It is also possible to use self-signed certificates.

Intrusion detection and breach reporting

The Compliant Kubernetes intrusion detection system is implemented as a daemon that monitors all syscalls and performs anomaly and intrusion detection. It is capable of detecting, among other things, the following types of intrusions:

- When a shell is running inside a container, e.g. someone using *kubectl exec*.
- When a container is running in privileged mode, or is mounting a sensitive path, such as */proc*, from the host.
- Unexpected reads of sensitive files, e.g. */etc/shadow*.
- When a standard system binary, e.g. *echo*, makes an outbound network connection.

Notably, the intrusion detection system logs suspicious activity to Elasticsearch. Any kind of integration with e.g. security operation centers are subject to custom inquiries and professional services.

The Service Provider is obliged to report any detected system breaches, outside of what the automatic intrusion detection system handles, by email to the customer.

Application logging

Log aggregation is performed for all containers in the Cluster. All container logs are stored in Elasticsearch and can be viewed in Kibana.

Limitations and exclusions:

- Logs are kept for a maximum of 30 days or up to 50 GB, whichever comes first.
- The service provider can setup long-term cold storage in an object storage, subject to professional services.

Audit logging

Audit logs for Compliant Kubernetes are stored in Elasticsearch. Customers may query this data themselves through Kibana. Audit logs are collected from the Kubernetes API-server with different levels of detail depending on the type of event and which lifecycle stage the event occured in.

Limitations and exclusions:

- Logs are kept for a maximum of 30 days or up to 50 GB, whichever comes first.
- The service provider can setup long-term cold storage in an object storage, subject to professional services.

Monitoring and alerting

Monitoring of the Cluster is done through the use of Prometheus and Grafana. Compliant Kubernetes is monitored and managed to ensure its availability and performance. Customers can view monitoring data through Prometheus, Grafana, and/or the provided Enterprise UI. The following are examples of metrics that are gathered:

- CPU usage per Pod and Node
- Memory usage per Pod and Node
- Disk I/O per Node

- Network I/O per Pod and Node

Customers may configure Alertmanager and Prometheus to send alerts based on any metrics available in Prometheus. The customer can set up alerts to various receivers, such as Slack, Email, or PagerDuty.

Detailed monitoring of customer Nodes and/or applications is out of scope. Customers may configure Prometheus to pick up more (application specific) metrics if desired.

Limitation and exclusions:

- Metrics are stored for a maximum of 30 days or up to 50 GB, whichever comes first.

Private container registry and vulnerability scanning

Compliant Kubernetes includes a private container registry. In this registry customers can store container images that should be used by Pods in the Cluster. Vulnerability scanning is performed on all container images uploaded in the private container registry. Admission policies can be enabled to prevent deployment of images with known vulnerabilities to the Cluster. These deployment time controls are complemented with runtime intrusion detection as described above.

Continuous integration and continuous deployment

Compliant Kubernetes is compatible with major CI/CD tools that support deploying to Kubernetes. Pre-configured CI/CD pipelines that use the private container registry and other features of Compliant Kubernetes are available subject to professional services. Service provider recommends GitLab and Jenkins as pipeline foundation.

Security policies

Compliant Kubernetes comes with a predefined set of policies for security and compliance. These are subject to change and are summarized below.

The following Pod security policies (PSPs) are set by default and cannot be overruled:

- No privileged containers are allowed and containers cannot run as root.
- Pods are not allowed to share the host IPC namespace.
- Pods are not allowed to share the host process ID namespace.
- Pods are not allowed to share the host network namespace.
- Pods are restricted to use only the following volume types:
  ConfigMap, EmptyDir, Projected, Secret, Downward API, and PersistentVolumeClaim.

Optional (best practices) policies that can be enabled upon request. Example policies include:

- Containers from other sources than the private registry are forbidden, e.g. it is impossible to run containers directly pulled from Docker Hub.
- All Pods must have some NetworkPolicy governing their network access. I.e. it is impossible to create a Pod outside of all NetworkPolicies.

Customer access and single sign-on

Compliant Kubernetes integrates with external authentication and authorization services such as SAML, LDAP, OpenID Connect, etc. Before Cluster initialization, customers select an external authentication service to be configured for the Cluster. Customers choose one account from the selected authentication service that receives customer administrator privileges. This includes administrator access to resources in the customers namespaces and view access to most resources in the Cluster. The customer administrator will also be able to grant other accounts privileges up to the same level.

Notably, customer administrators do not have “cluster-admin” privileges in the Cluster and may, e.g., not disable audit logging, turn off intrusion detection, or modify certain security policies. Such changes are subject to consultation and professional services.

4. Operations

The following section describes the operational tasks that are performed for the managed Compliant Kubernetes Service.

Backup and disaster recovery

Backups of Compliant Kubernetes are performed for the purpose of disaster recovery. The backup scope includes:

- All gathered monitoring data.
- Audit logs for the Kubernetes API server.
- Application (container) logs from Cluster Services and key External Services.
- The subset of Kubernetes Cluster state required to restore Cluster Services and External Services.
- Customer Kubernetes resources including Pods, deployments, stateful sets, daemonsets, cron jobs, Services, horizontal Pod autoscalers, Pod disruption budgets, configmaps, secrets, network policies, Service accounts, roles, role bindings, ingresses, persistent volumes, and custom resources.

Backup frequency: once per day, to be performed between 0:00am and 3:00am CET.
Number of backups kept: 3
Backup destination: cloud provider object storage

Disaster recovery is committed to be completed within 4 hours. Some applications may need manual intervention after a recovery in order to become fully operational, e.g. applications that require specific initialization or depend on other components being available.

The following limitations apply:

- Customers cannot access backup configurations and raw backup.
- Only data required to run Compliant Kubernetes is covered.
- Customers are responsible to backup any kind of user and application data beyond what is covered by the Compliant Kubernetes resources listed above (the backup scope).

Updates and upgrades

All planned updates are performed in the maintenance window that occurs Wednesdays, 7.00am - 12.00am (CET). For planned updates, we distinguish between:

- *Minor updates*. These are performed at most every Wednesday. Minor updates cause no downtime to customer applications (if these tolerate Node replacement, see below) nor any downtime to the control plane, Cluster Services or External Services. Minor updates preserve backwards compatibility of any APIs of Kubernetes, Cluster Services, and External Services.
- *Major updates.* These are performed at most once per month (first Wednesday of the month), in order to provide a smooth upgrade experience according to the Kubernetes version skew policy. Major updates cause no downtime to customer applications (beyond Node replacement, see below), but may cause downtime in the Kubernetes control plane, Cluster Services, and/or External Services. Any downtime will occur in the maintenance window exclusively, and will always be shorter than 1 hour. Major updates cause no loss of logging data during potential downtime.
  If a major update requires customer action such as updating Kubernetes API objects, this will be announced 2 weeks in advance, and the Service Provider will include a migration guide for all affected objects. Upgrades of objects can be performed on customer Clusters directly, without any downtime expected. Should customers want to test upgrades to a non-production environment, the Service Provider can, subject to professional services, create a staging Cluster with the next major version.

Unplanned updates include critical security patches and Cluster recovery upon force majeure situations such as major power outages, natural disasters, and the similar.
Unplanned updates can occur outside the maintenance window and will be notified as far in advance as possible.

Planned updates include upgrades to more recent versions of Kubernetes, Cluster Services, External Services as well as other components of Compliant Kubernetes.
The following specifies the target upgrade cadency:

Major and minor Kubernetes release upgrades:

- Upgrades will occur when the Service Provider has ensured compatibility with provided Services, and will be performed at most weekly (minor updates) and monthly (major updates), respectively.
- Customers can postpone upgrades for up to 1 month (major update frequency).

Kubernetes patch release upgrades:

- Included. Upgrades will occur when the Service Provider has ensured compatibility with provided Services.

  VM OS:
- Yearly, but may occur more frequently depending on the severity of the vulnerabilities that need patching.

Other components:

- No time commitment for upgrades. The Service Provider chooses appropriate setup and upgrade timing.

Customer applications are expected to tolerate worker Node replacement of one Node at a time (adding a new Node, draining the old and removing it) to be able to continue normal operations throughout normal upgrades, updates, and patches.

5. Service Level Objective (SLO)

Service provider shall use commercially reasonable efforts to make the Cluster, each Cluster Service and External Service available 24 hours a day, 7 days a week, with an overall **99.95**% annual availability for your Cluster (i.e. 365 days minus 4h20min), except for:

- Planned downtime and maintenance events;
- Force Majeure Events;
- Unavailability of the External Services and Enterprise UI (whose SLO is 99.5%);
- Failures or malfunctions in any Client software, equipment or technology; and/or
- Downtime due to Node replacement (Customer applications are expected to tolerate worker Node replacement of one Node at a time).

Health is measured every minute for all the above Services. The uptime is then calculated separately for each Service as the number of healthy responses divided by the total number of measurements during the period.

Disclaimer: Customers shall not interfere with or disrupt the integrity or performance of the Managed Service.

6. Service requests

Service provider is obliged to fulfill the following actions upon request through the Service Desk:

- Provision a new Cluster.
- Scale Cluster out or in by adding or removing Nodes.
- Scale Cluster up or down by resizing Nodes.
- Increase storage for Cluster Services or External Services.
- Decommission Cluster.
- Handle incidents: investigate, classify, and resolve whenever applicable.

Service desk contact information:

- Email: support@elastisys.com
- Slack: shared channel: elastisys.slack.com
- Telephone: +46 730 914 802

Target response time for service requests is 1 hour, except for provisioning of a new Cluster, which is performed within 1 business day after the customer submitted a completed onboarding form (see Section “Onboarding form”). Service requests are handled during 8 am to 5 pm CET. Incidents (see below) are handled 24/7 with a response time of 1 hour.

Cloud native professional services and training

Being a Kubernetes Certified Service Partner (KCSP) and a Kubernetes Training Partner (KTP), the Service Provider can help customers throughout their whole cloud native transformation, including training, architecture guidelines, containerization, implementation of CI/CD, compliance using cloud native tooling, DevOps, DevSecOps, etc. For Compliant Kubernetes in particular, the professional services offer allows customers to gain the most out of the platform and use Kubernetes as well as Cluster Services and External Services to their full potential.

Minimum infrastructure requirements

Minimum Compliant Kubernetes IaaS requirements on Exoscale	
Environment	Instance type
Workload cluster (Customer)	
Master 1	medium
Master 2	medium
Master 3	medium
Service Cluster	
Master	medium
Master	medium
Master	medium
Worker ES1	large
Worker ES2	large
Worker ES3	large
Worker InfluxDB	large
Worker rest 1	large
Worker rest 2	large
Additional requirements	
Feature	Number
Elastic IP for the ingress in the service cluster	1
Local discs for each service cluster VMs (50 GBs)	12
Object Storage	500 GB

7. Division of responsibility for the managed service

Responsibility assignment matrix; Responsible, Accountable, Consulted, Informed (RACI).

Setup and contributions

ID	Activity	Customer				Service Provider			
		R	A	C	I	R	A	C	I
	Definition of Compliant Kubernetes Architecture				X	X	X		
	Contribution of all related software components needed to run Compliant Kubernetes, including Cluster Services and External Services					X	X		
	Contribution of all related software licenses needed to run Compliant Kubernetes					X	X		
	Installation and configuration of all related Compliant Kubernetes components: ● Setup of virtual machines and related infrastructure on the cloud provider ● Setup of Compliant Kubernetes ● Setup of related networking configuration (accessible on public Internet IP) ● Setup of initial user privileges Notes: Customer selects Cluster dimensioning as well as authentication provider for installation.			X	X	X	X		
	IaaS costs Notes: IaaS costs for Compliant Kubernetes Cluster (including Cluster Services and External Services), as well as any additional costs associated with performed Service Desk requests are the responsibility of the customer.	X	X						

Maintenance and operations

ID	**Activity**	**Customer**				**Service Provider**			
		R	A	C	I	R	A	C	I
	Administration of relevant user privileges Notes: Connection to external authentication provider is the customer's responsibility			X	X	X	X		
	Planned major updates and unplanned updates			X	X	X	X		
	Updating customer Kubernetes objects (resources) as required for major updates.	X	X					X	
	Planned minor upgrades				X	X	X		
	Monitoring of key metrics (CPU, RAM, disk space)				X	X	X		
	Adding/deleting/starting/stopping compute Nodes during maintenance					X	X		
	Performing weekly maintenance work in maintenance window					X	X		
	Backup				X	X	X		
	Recovery				X	X	X		
	Responsibility for any kind of customer application	X	X						
	Aggregation of all container and audit logs					X	X		
	Ensure that application tolerates node replacement	X	X						

Decommissioning

ID	**Activity**	**Customer**				**Service Provider**			
		R	A	C	I	R	A	C	I
	Extraction of customer data and application	X	X						
	Shutdown and removal of Compliant Kubernetes application elements including data					X	X		
	Decommissioning of IaaS infrastructure					X	X		

Performance management

ID	Activity	Customer				Service Provider			
		R	A	C	I	R	A	C	I
	Collecting performance metrics					X	X		
	Performing configuration changes that affects Cluster capacity, including: ● Scale Cluster up or down ● Scale Cluster in or out ● Modify storage capacity for logs increase (i.e. switch to larger underlying hosts, or increase storage for logs) Notes: Via Service Desk. Customer is responsible to ensure sufficient capacity upon scale in or down.					X	X		
	End-to-end performance of customer application(s) Notes: Performance diagnosis subject to professional services.	X	X					X	

Incident management

ID	Activity	Customer				Service Provider			
		R	A	C	I	R	A	C	I
	Investigation and classification of incidents			X		X	X		
	Performing configuration changes Notes: Subject to professional services. The Service Provider may refuse changes (without justification) if they may limit system maintainability.			X		X	X		
	Setup and maintenance of ticketing system					X	X		
	Documentation of recovery				X	X	X		

Target response time for incidents is 1 hour, 24/7.

8. Onboarding form

The following form needs to be filled in for provisioning a new Cluster. Once the form is completely filled out and submitted to the Service Provider, a Cluster is created within at most 1 business day.

How to send this form:

- Complete the form either as PDF or text file, so as to make it easy to copy-paste.
- Encrypt the file, with **either**:
  - Compress the form with encryption as a ZIP file.
  - Use GPG: gpg --symmetric --armor form.txt
- Send the encrypted file per email.
- Send the password to the encrypted file either via Slack or SMS.

Compliant Kubernetes Onboarding Form v2020-03-11	
Customer (e.g., “Customer GmbH”)	
Desired DNS_LABEL (e.g., “customer-gmbh”) This will be used in the URLs for your Services, e.g. “grafana.customer-gmbh.a1ck.io”.	
Cloud Provider	☐ Exoscale If you want to setup a cluster in an existing organization, then fill out the API key and secret below. Otherwise, leave the next rows empty.
Endpoint (e.g., "https://api.exoscale.com/v1")	
API Key (e.g., “EXO…”)	
API Secret	

Identity Provider SAML	☐ Enable
IDP Callback URL is configured to: https://dex.DNS_LABEL.a1ck.io/callback	☐ Yes ☐ No (please contact support)
IDP is configure with: ● user attribute "user" ● email attribute "email" ● groups attribute "groups"	☐ Yes ☐ No (please contact support)
URL (e.g., "https://idp.example.com")	
CA File URL (e.g., "https://idp.example.com/ca.pem")	
Identity Provider OpenID	☐ Enable
IDP Callback URL is configured to: https://dex.DNS_LABEL.a1ck.io/callback	☐ Yes ☐ No (please contact support)
Issuer discovery URL (e.g., "https://accounts.example.com")	
Client ID	
Client Secret	
Another Identity Provider	☐ Enable (please contact support)