Managed Kubernetes
Terms of Service

Last updated on: October 13, 2020

Contents

1. General provisions

1.1. Composition of the Service

These terms of the «Managed Kubernetes» service (Terms of service) contain information that effectively discloses the purpose and composition of the Managed Kubernetes Service (hereinafter referred to as the Service), which constitutes a set of technical services continuously provided by the Contractor in the form of a subscription service, including the following:

- Installing Kubernetes (https://kubernetes.io/) cluster(s) according to the parameters defined in the Request based on the computing infrastructure provided by the Customer that meets the requirements specified in these Terms of Service.
- Installing auxiliary software components that are deeply integrated into the clusters being installed and implementing an infrastructure environment suitable for the operation of Customer's applications and various software (hereinafter referred to as Software).
- The post-installation 24x7x365 technical support of clusters and auxiliary software components, including regular updates using the latest versions of clusters and auxiliary components developed by the Contractor including monitoring systems and components for aggregating operating parameters of the cluster.
- Round-the-clock monitoring of the health of Kubernetes clusters and auxiliary components based on Prometheus and Grafana, as well as responding to failures and errors in their operation in accordance with the rules defined in the Regulations.
- Systematic backup of the configuration of the cluster's control components to Customer's or Contractor's storage (at the discretion of the Customer).
- Consulting support of Contact persons of the Customer to the extent and under the rules set out in these Terms of Service.
- Prompt error fixing, developing, upgrading, and adapting auxiliary components of Kubernetes clusters carried out by engineers of the Contractor in accordance with their own production plans and internal technical regulations or as agreed upon by the Parties.

1.2. Types of clusters and nodes

The following types of cluster nodes are supported:

- **Cloud** node is a virtual machine (or a dedicated server) that is created by Kubernetes components fully automatically. These nodes can be re-created (using the RollingUpdate strategy) if there is a need to make changes.
- **VM** node is a virtual machine pre-created or created manually due to a lack of automation capabilities (no API, non-standard/unsupported API, virtualization platform,

or provider - see paragraph 1.6). The Contractor must have access to controls (to view the status of the VM, reboot it, re-install the guest operating system) via a web browser, or SSH, or by contacting the support service.

- **Hard** node is a dedicated server. The Contractor must have access to controls (to view the status of the server, reboot it, reinstall the operating system) via a web browser or by contacting the support service.
- **Special** node is any node that requires special treatment and cannot be removed from the cluster and/or reinstalled (rescheduled) without preparatory activities and/or separate coordination. This type of nodes can, for example, include stateful nodes (which store data locally), well-known nodes (their addresses are used for connection), session-termination ones (they have permanent connections that cannot be closed and re-established correctly), nodes with the GPU computing support, and nodes running calculations that cannot be interrupted.

Depending on the cluster architecture, the operating environment, and the type of nodes used, the following three **types of clusters** are supported:

#	Cluster type	Description	Requirements and limitations	Allowed types of nodes
1.2.1.	**Cloud**	The Kubernetes cluster is hosted on one of the supported cloud providers or a private cloud.	- All cluster nodes (workers and masters) are located in the same region of the cloud provider.	- Cloud - Special
1.2.2.	**Non-Cloud**	The Kubernetes cluster is hosted on pre-created VMs and/or dedicated servers.	- All cluster nodes (workers and masters) are located in the same data center; - There is a unified and stable L2 network that connects all cluster nodes; - The functionality of Container Storage Interface (CSI) and the ability to create LoadBalancer services may be limited or not available.	- VM - Hard - Special
1.2.3.	**Hybrid**	The Kubernetes cluster is hosted on pre-created VMs and/or dedicated servers; however, some nodes are created automatically in one of the supported cloud providers or a private cloud.	- All cluster nodes (workers and masters) are located in the same data center; - There is a unified and stable L2 network that connects all cluster nodes; - The functionality of Container Storage Interface (CSI) and the ability to create LoadBalancer services may be limited or not available.	- Cloud - VM - Hard - Special

The **maximum number of nodes** (including master nodes) per cluster of any type is **150** (including up to 3 master nodes). This limitation is primarily due to the inability to ensure the stable collection and processing of metrics at required intervals in clusters with more than 150 nodes due to Prometheus limitations. In the case of higher amounts of nodes, we recommend splitting them into several clusters.

1.3.   Tariff plans

1.3.1.   General information

##		Standard	Silver	Gold	Platinum
1.3.1.1.	**Purpose**	Development environments and non-critical production environments	Production environments	Critical production environments	Critical production environments with a high service level
1.3.1.2.	**Relative overall availability level**	Low	Average	High	
1.3.1.3.	**The minimum subscription period**	No			6 months
1.3.1.4.	**Available Kubernetes versions**	The latest two 1.17, 1.18	The latest four 1.15, 1.16, 1.17, 1.18		
1.3.1.5.	**The maximum number of nodes (per type) in a cluster**	15 Hard / 15 Special / 30 VM / 60 Cloud	150		

1.3.2.   Support for advanced components

#	Service Type	Standard	Silver	Gold	Platinum
1.3.2.1.	GPU support Special edition of Docker	No	Yes	Yes	Yes
1.3.2.2.	Istio Support	No[1]	No[1]	Yes[2]	Yes[2]

[1] Istio can be used in a development cluster regardless of the selected plan if you are already using Istio in a production cluster (with a plan where Istio is available).

[2] Expected in the IV quarter of 2020. Additional fees apply.

1.4.   Regular business hours of the Contractor

During regular business hours the Contractor performs:

- planned tasks;
- automatic updates;
- significant updates.

Regular business hours of the Contractor means 10:00 to 18:00 (GMT+3) Monday through Friday except on legal holidays.

Non-business hours of the Contractor means all the time that is outside regular business hours.

1.5.   Technical support

#	Service Type	Standard	Silver	Gold	Platinum
1.5.1.	Response time[1] to incidents	up to 15 minutes	up to 10 minutes	up to 5 minutes	up to 5 minutes
1.5.2.	Initial installation and configuration of the cluster	Yes			
1.5.3.	24x7x365 monitoring and handling cluster incidents	Yes			
1.5.4.	Backup of essential components of the platform (etcd)	Yes			
1.5.5.	Technical advice on cluster architecture and applications (during regular business hours)	Yes			
1.5.6.	Regular conference calls To synchronize joint activities, discuss project progress, identify needs/issues/goals. Conducted by an implementation engineer.	No		Yes, once per month	Yes, once per month
1.5.7.	Voice consultations with lead engineers during regular business hours	No		Yes	Yes
1.5.8.	Dedicated technical contact (lead engineer)	No			Yes
1.5.9.	Delivery of the Service when the Contractor does not have direct SSH access to the cluster	No			Yes

1.5.10.	Conducting an audit by information security service (ISS) Involves collective conference calls with the Customer's ISS specialists, discussing system design and practices in use, evaluating results of the audit conducted by the Customer's ISS.	Direct access means direct SSH access via a public IP address (Contractor's IP address pool can be limited if necessary) or access via a site-to-site VPN. No	Yes, Up to 3 hours	Yes, Up tp 5 hours
1.5.11.	Urgent (prioritized) error fixing in upstream components Fixing errors in standard Open Source components of the Kubernetes cluster, including control-plane, cloud-providers, Prometheus, and others.	No		Yes, charged separately (see paragraphs 5.2 and 5.4)
1.5.12.	Deep customization and additional features implementation	No		Yes, charged separately (see p. 5.2.)

[1] Refer to the p. 1 of the Agreement for more information about the Reaction time and communication process in general.

1.6. Supported location options

#	Type of infrastructure	Cluster autoscaling	CSI (disk subsystem)	LoadBalancer (service of the LoadBalancer type)	Minimum plan
Location options for Cloud clusters					
1.6.1.	Amazon AWS	Yes	Yes	Yes	Standard
1.6.2.	Google Cloud Engine	Yes	Yes	Yes	Standard
1.6.3.	OVH Cloud	Yes	Yes	Not guaranteed	Standard
1.6.4.	OpenStack	Yes	Not guaranteed	Not guaranteed	Silver
1.6.5.	VMware vSphere[1]	Yes	Not guaranteed	Not guaranteed	Platinum
Location options for Non-Cloud clusters					
1.6.6.	Bare-metal	No	Not guaranteed	Not guaranteed	Standard
1.6.7.	Virtual machines	No	Not guaranteed	Not guaranteed	Standard
Location options for Hybrid clusters					
1.6.8.	OVH placement	Yes	Yes,	Not guaranteed	Standard

				only Cloud nodes		

2.   Updating clusters

The Kubernetes cluster and its components are constantly evolving, with functional and security issues being detected and fixed during operation. Also, new features emerge regularly. That's why Managed Kubernetes from Flant is systematically tested and updated.

2.1.   Types of updates

The following types of updates are performed as part of the Service:

- **Significant update** involves updating the minor version of Kubernetes. Examples include updating Kubernetes from 1.15 to 1.16 (but not patch versions, i.e. not from 1.15.2 to 1.15.3), updating Ingress controller, as well as updates requiring eviction of nodes (the maintenance mode) and any other updates that lead to downtime of applications running in the cluster.
- **Automatic update** involves a large-scale automatic update of cluster components monitored by an engineer.

Depending on their composition, updates are divided into two groups:

- **Release** — a planned update of cluster components dedicated to extending or changing functionality as well as installing a set of patches to fix errors.
- **Hotfix** — an unplanned update of cluster components to fix critical errors or other functional issues that have already led (or may lead) to the instability of the cluster operation.

Single Releases or Hotfixes are always released as automatic updates. Significant updates may involve the simultaneous installation of several Releases and/or Hotfixes in addition to updating the Kubernetes version.

2.1.1.   Update channels

The Contractor conducts comprehensive testing of new Releases and Hotfixes, including unit and integration testing of individual parts of the system, the end-to-end testing of clusters in lifelike conditions, manual verification of significant changes. At the same time, best practices in SRE require using approaches that reduce the potential negative impact of the problem that was not detected during the testing.

There are various requirements to clusters as a part of infrastructure. Production clusters (unlike development ones) have more strict and higher requirements for reliability. It is best to reduce

the frequency of updates and limit changes to truly necessary ones. Components should be tested as much as possible. However, the low rate of changes in the cluster components will result in delays with implementing new features.

That's why we offer five **update channels** for cluster components so the optimum balance between the rate of change and reliability can be achieved. Updates grouped as Releases or Hotfixes are sequentially applied to clusters for each update channel, starting with the least stable one, Alpha, and ending with the most safe and reliable one, RockSolid. Thus, the Alpha update channel has the highest rate of emergence of new features and frequency of updates while being the least stable and secure. The RockSolid update channel, in turn, has the lowest rate of emergence of new features and frequency of updates while providing the highest stability. The more or less uniform distribution of all maintained Kubernetes clusters between update channels implies that updates intended for the RockSolid channel have been successfully applied to at least 80% of clusters within other update channels.

#	Update channel	Description	Standard time of updates	Requirements to versions in the channel	When the user is notified about the changes
2.1.1.1.	**Alpha**	The least stable versions	No restrictions	No restrictions	At the time of the update
2.1.1.2.	**Beta**	Stable versions that have been running successfully on some clusters maintained by the Contractor	No restrictions	1 day of normal operation in the **Alpha** update channel for **releases**. 2 hours of normal operation in the **Alpha** update channel for **hotfixes**.	At the time of the update
2.1.1.3.	**EarlyAccess**	Stable versions tested on at least 10% of clusters maintained by the Contractor.	Thu, 12:00-13:00 (GMT+3) for **releases**. No restrictions for **hotfixes**.	1 day of normal operation in the **Beta** update channel for **releases**. 2 hours of normal operation in the **Beta** update channel for **hotfixes**.	At least 2 hours before the update

2.1.1.4.	**Stable**	Stable versions tested on at least 40% of clusters maintained by the Contractor.	Wed, 12:00-13:00 (GMT+3) for **releases**. Regular business days, 14:00-15:00 (GMT+3) for **hotfixes**.	5 days of normal operation in the **EarlyAccess** update channel for **releases**. 1 day of normal operation in the **EarlyAccess** update channel for **hotfixes**.	At least 24 hours before the update
2.1.1.5.	**RockSolid**	Stable versions tested on at least 80% of clusters maintained by the Contractor.	Tue, 12:00-13:00 (GMT+3) for **releases**. Regular business days, 14:00-15:00 (GMT+3) for **hotfixes**.	12 days of normal operation in the **Stable** update channel for **releases**. 5 days of normal operation in the **Stable** update channel for **hotfixes**.	At least 7 days before the update

2.2. Parameters of cluster updates

#	Option	Standard	Silver	Gold	Platinum
2.2.1.	The timing of automatic updates	During regular business hours (p. 1.4.)			
2.2.2.	Selecting a certain time window for automatic updates	No	Yes		
2.2.3.	Performing automatic updates outside regular business hours	No	Only for the **RockSolid** update channel[1]		
2.2.4.	The timing of significant updates	At the agreed interval during regular business hours (p. 1.4.)			
2.2.5.	Performing significant updates and other tasks (by engineers) outside regular business hours	No	Yes, for an additional charge (see p. 5.1)		
2.2.6.	Possibility to use the EarlyAccess update channel	Yes			
2.2.7.	Possibility to use the Stable update channel	No	Yes		

2.2.8.	Cost of using the RockSolid update channel	Not applicable	Free of charge for 1/3 of clusters[2] if their total amount exceeds 3; in all other cases — 400 \$/cluster	Free of charge for 1/3 of clusters[3] if their total amount exceeds 3; in all other cases — 200 \$/cluster	Free of charge, but no more than 1/2 of all clusters[4]

[1] Any system changes can potentially lead to errors. That is also true for changes made outside regular business hours. While all updates are monitored by an engineer, emergencies may require the participation of leading engineers and system architects. With this in mind, updates outside regular business hours are only available for the most stable RockSolid update channel since it contains modifications that are guaranteed to operate smoothly on at least 80% of clusters maintained by the Contractor.

[2] for Silver clusters

[3] for Gold clusters

[4] for Platinum clusters

3. Service level agreement (SLA)

This service level agreement establishes conditions and a method for measuring the level of service at a given time, defines the guaranteed level of service (see paragraph 3.2), and sets out penalties for violating the guaranteed level of service (see paragraph 3.3).

The level of service is measured based on monitoring data for individual cluster components (and not for the cluster as a whole) grouped into availability types (see p. 3.1). Some availability types require enabling a special high availability mode of operation of components (modules) of the Kubernetes cluster (hereinafter referred to as the **High Availability** mode, **HA**) to ensure a guaranteed level of service.

Specificities and operating conditions of components running in the **High Availability** mode:

- The component is running in two (or more) instances to preserve availability if a single instance fails.
- Enabling this mode requires additional computing resources.
- If there is more than one master node in the cluster, the High Availability mode is activated for all components by default.
- The HA mode can be turned off for individual components (such as authentication, monitoring, etc.), thus reducing the cost of computing resources. However, we cannot guarantee the Service Level (SL) for such subsystems.
- Control Plane, the cluster core, is considered running in the HA mode if it has at least three instances.

3.1.   Availability types

To accurately identify types of degradations and the degree of their impact on the performance of services, we define several types of unavailability for which it is possible to set the parameters under the service level agreement (SLA).

3.1.1.   Synthetic Availability

The availability of (pre)running applications in the cluster and cluster components that are not related to the availability of the cluster's Control Plane, such as:

- applications that have been started before (in containers that are still running);
- Kubernetes services (the presence of corresponding iptables rules);
- DNS (the ability to resolve hosts).

The SLA for Synthetic Availability is valid only if the basic conditions are met:

- cluster nodes continue to be available;
- there is network connectivity between nodes;
- the application continues to operate correctly.

At the same time, applications keep on running if they meet the following conditions:

- the application itself caches DNS data;
- the application can operate with no scheduler or autoscaler running as well as with no ability to add/edit service endpoints;
- no kube-apiserver availability is required for the application to operate.

This availability type **does not require** enabling the **HA** mode for components.

3.1.2.   Nginx Availability

The availability of Ingress Nginx and its components, which means that incoming HTTP requests are delivered to end applications.

The Nginx Availability SLA does not cover the network availability that depends on the infrastructure.

This availability type **requires** enabling the **HA** mode for components.

3.1.3.   Node Group Availability

In Managed Kubernetes, nodes are tied together in groups based on the purpose and managed as an integrated entity. It means that all nodes in the group have the same metadata defined in the

parameters of a NodeGroup resource based on labels, annotations, or taints fields. Such a group of nodes is called the Node Group.

Node Group Availability indicates that each Node Group has at least N-1 correctly functioning nodes, where N is the minimum number of nodes in NodeGroup as defined in the configuration.

The Node Group Availability SLA does not apply to cases when a node fails due to improper operation of an application installed by the customer (e.g., due to memory leaks).

This availability type **does not require** enabling the **HA** mode for components.

3.1.4.   Control Plane Availability

This availability type indicates that the deployment process in the cluster is working, and self-healing processes are running. It formulates the availability level of the main control Kubernetes components:

- etcd;
- kube-apiserver;
- kube-controller-manager;
- kube-scheduler;
- kube-dns;
- other components that the API depends on (such as vertical-pod-autoscaler's admission controller, cert-manager's cainjector, etc.).

This availability type **requires** enabling the **HA** mode for components.

3.1.5.   M&A Availability

This availability type means that monitoring and autoscaling are operating correctly. It defines the availability level of Prometheus and the autoscaling functionality, namely, of components and data sources:

- Prometheus (the operation of the TSDB and alerting);
- Trickster;
- prometheus-metrics-adapter;
- vertical-pod-autoscaler;
- horizontal-pod-autoscaler;
- key data sources, such as:
  - kube-state-metrics;
  - ingress-nginx;
  - node-exporters and kubelet (at available nodes).

This availability type **requires** enabling the **HA** mode for components.

3.1.6. Extensions Availability

This availability type means that additional Kubernetes components are available. It sets out the availability level of components not directly involved in delivering applications to the runtime environment and their operation in the cluster:

- Dex (providing SSO authentication in the cluster);
- Grafana;
- Dashboard;
- longterm-prometheus;
- cluster-autoscaler;
- cloud-controller-manager;
- machine-controller-manager (the ability to order new nodes);
- openvpn.

This availability type **requires** enabling the **HA** mode for components.

3.1.7. General SLA conditions for all availability types

3.1.7.1. Any availability type does not apply to situations of complete lack of network connectivity between nodes or total inaccessibility of the cluster infrastructure.

3.1.7.2. The availability of each cluster component individually and of the availability type as a whole is determined by the best value obtained from multiple probing agents every 30 seconds.

3.2. Guaranteed level of service

The guaranteed service level depends on the availability type and the chosen update channel under the SLA.

#	Availability type	Standard	Silver	Gold	Platinum
3.2.1.	Synthetic Availability	99,90%	- **99,95%**, if the **Stable** or **RockSolid** update channel is used - **99,90%**, if the **EarlyAccess** update channel is used	- **99,99%**, if the **RockSolid** update channel is used - **99,95%**, if the **Stable** update channel is used - **99,90%**, if the **EarlyAccess** update channel is used	

3.2.2.	Nginx Availability	• **99,90%**, if the **HA** mode is enabled and dedicated nodes are used • **SL is not guaranteed** in all other cases (less than 2 dedicated nodes)	• **99,95%**, if the **HA** mode is enabled, dedicated nodes are used, and the **Stable** or **RockSolid** update channel is used • **99,90%**, if the **HA** mode is enabled, dedicated nodes are used, and the **EarlyAccess** update channel is used • **SL is not guaranteed** in all other cases (less than 2 dedicated nodes)	• **99,99%**, if the **HA** mode is enabled, dedicated nodes are used, and the **RockSolid** update channel is used • **99,95%**, if the **HA** mode is enabled, dedicated nodes are used, and the **Stable** update channel is used • **99,90%**, if the **HA** mode is enabled, dedicated nodes are used, and the **EarlyAccess** update channel is used • **SL is not guaranteed** in all other cases (less than 2 dedicated nodes)
3.2.3.	Node Group Availability	**99,50%**	• **99,90%**, if the **Stable** or **RockSolid** update channel is used • **99,50%**, if the **EarlyAccess** update channel is used	• **99,95%**, if the **RockSolid** update channel is used • **99,90%**, if the **Stable** update channel is used • **99,50%**, if the **EarlyAccess** update channel is used
3.2.4.	Control Plane Availability	• **99,50%**, if 3 masters are running on dedicated nodes • **SL is not guaranteed** in all other cases (less than 3 masters)	• **99,90%**, if 3 masters are running on dedicated nodes and the **Stable** or **RockSolid** update channel is used • **99,50%**, if 3 masters are running on dedicated nodes and the **EarlyAccess** update channel is used • **SL is not guaranteed** in all other cases (less than 3 masters)	• **99,95%**, if 3 masters are running on dedicated nodes and the **RockSolid** update channel is used • **99,90%,** if 3 masters are running on dedicated nodes and the **Stable** update channel is used • **99,50%**, if 3 masters are running on dedicated nodes and the **EarlyAccess** update channel is used • **SL is not guaranteed** in all other cases (less than 3 masters)
3.2.5.	M&A Availability	• **99,50%**, if dedicated nodes are used and the HA mode is enabled • **SL is not guaranteed** in all other cases	• **99,50%**, if dedicated nodes are used and the **HA** mode is enabled • **SL is not guaranteed** in all other cases	• **99,95%,** if dedicated nodes are used, the **HA** mode is enabled, and the **RockSolid** update channel is used • **99,50%**, if dedicated nodes are used and the **HA** mode is enabled • **SL is not guaranteed** in all other cases
3.2.6.	Extensions Availability	• **99,00%**, if dedicated nodes are used and the HA mode is enabled • **SL is not guaranteed in all other cases**	• **99,50%**, if the HA mode is enabled, dedicated nodes are used, and the **Stable** or **RockSolid** update channel is used • **99,00%**, if the HA mode is enabled, dedicated nodes are used, and the **EarlyAccess** update channel is used • **SL is not guaranteed** in all other cases	• **99,50%**, if dedicated nodes are used, the HA mode is enabled, and the **Stable** or **RockSolid** update channel is used • **99,00%**, if dedicated nodes are used, the **HA** mode is enabled, and the **EarlyAccess** update channel is used • **SL is not guaranteed** in all other cases

Hereinafter, the dedicated node means a cluster node that performs a single functional task (excluding the mandatory system tasks). Examples of such nodes include a dedicated frontend node (where Ingress controller is running), a dedicated master node (where only control plane components are running), a dedicated monitoring node (where only monitoring-related applications are running), etc.

3.3. Penalties for violating the service level

If the SLA is violated for multiple availability types for the same cluster simultaneously, only the highest penalty for the availability type is taken into account.

#	Penalty	Standard	Silver	Gold	Platinum
3.3.1.	The maximum amount of penalty per month of service for violating the SLA[1]	**No more than 25%** of the monthly cost of service	**No more than 50%** of the monthly cost of service	**No more than 75%** of the monthly cost of service	
3.3.2.	Amount for 1 hour of downtime	**1%** of the total monthly maintenance cost for this cluster	**3%** of the total monthly maintenance cost for this cluster	**5%** of the total monthly maintenance cost for this cluster	

[1] Excluding downtime due to circumstances beyond the Contractor's control, such as infrastructure provider's downtime, problems with payment to the infrastructure provider, force majeure, etc.

4. Pricing

4.1. Cluster maintenance

#	Name	Standard	Silver	Gold	Platinum
4.1.1.	Basic price of a Cloud cluster[1], $/month	500			
4.1.2.	Basic price of a Non-Cloud cluster[1], $/month	700			
4.1.3.	Basic price of a Hybrid cluster[1], $/month	700			
4.1.4.	Contact persons on the Customer's side[2], $/month	300			
4.1.5.	Additional contact persons on the Customer's side, the surcharge for each additional contact person, $/month	200			
4.1.6.	Service level surcharge for the Cloud cluster, $/month[3]	0	200	800	1600
4.1.7.	Service level surcharge for the Non-Cloud cluster, $/month[3]	0	400	1000	1800
4.1.8.	Service level surcharge for the Hybrid cluster, $/month[3]	0	400	1000	1800
4.1.9.	The surcharge for the higher service level or an unlimited plan (any number of clusters, any cluster type)[4], $/month	-	3000	5000	8000

[1] The basic price of cluster maintenance includes the maintenance of worker nodes for up to 200 $ and no more than 3 dedicated master nodes.

[2] The mandatory option for the first cluster of the Customer; includes up to two contact persons. Contact persons are not tied to a specific cluster. This option is not mandatory for the second and subsequent clusters.

[3] If a redundancy (either hot or cold) is required, the service level surcharge for an additional cluster in an identical infrastructure is not charged.

[4] Provided that no more than two different infrastructure types are used for clusters, including different types of clusters and cloud providers in the case of a Cloud cluster.

4.2.   Maintaining cluster nodes

#	Type	Standard	Silver	Gold	Platinum
4.2.1.	Cloud node, $/month	10			
4.2.2.	VM node, $/month	20			
4.2.3.	Hard node, $/month	40			
4.2.4.	Special node, $/month	100			

5.   Additional services

#	Type	Standard	Silver	Gold	Platinum
5.1.	Performing significant updates during regular business hours	No	200 $ per hour, a minimum of 3 hours		
5.2.	Performing additional works during regular business hours	160 $ per hour, a minimum of 2 hours			
5.3.	Performing additional works outside of business hours	320 $ per hour, a minimum of 3 hours			
5.4.	Developing, adapting, refining, and correcting the upstream versions of Kubernetes and other software of the Kubernetes ecosystem	120 $ per hour, a minimum of 10 hours			