

# GIGA CLOUD

## СaaS (Container as a Service)

### Посібник користувача

версія 1.9 05\_2020

03022, Україна, м. Київ,  
вул. Васильківська, 37-В

+38 (044) 233-71-70  
sales@gigacloud.ua

[www.gigacloud.ua](http://www.gigacloud.ua)

## Зміст

1. Призначення та головні параметри сервісу CaaS у хмарі GigaCloud .....	3
2. Початок користування послугою з метою тестування .....	3
2.1. Замовлення тестової конфігурації .....	3
2.2. Доступ до ресурсів кластеру (клієнтська частина) .....	6
Ubuntu, Debian .....	6
Centos, RHEL .....	6
macOS .....	6
Windows .....	6
2.3. Управління інфраструктурою .....	7
2.4. Приклад створення робочої інфраструктури .....	8
3. Поширені запитання і відповіді .....	11
Як додати новий пул до вже існуючого кластеру? .....	11
Додатковий функціонал підключення «білих» (публічних) IP адрес .....	11
Як дізнатися про робочі навантаження нод кластеру? .....	12
Як деактивувати тестову інфраструктуру та видалити непотрібні ноди кластеру? .....	12

## 1. Призначення та головні параметри сервісу CaaS у хмарі GigaCloud

**CaaS (Container as a Service)** – це послуга надання через Інтернет в оренду хмарних ресурсів провайдера, з розгортанням **Kubernetes-кластеру в режимі самообслуговування**.

Ваш кластер (сукупність пов'язаних обчислювальних вузлів) має наступні ключові властивості:

- містить одну **мастер-ноду** (керуючу віртуальну машину, сервер);
- кожний пул (набір однотипних вузлів) у складі кластеру містить якнайменше, одну **робочу ноду** (віртуальну машину, сервер для запуску подів з контейнерами);
- **пул може масштабуватися** в хмарі GigaCloud у разі потреби, в обидва боки. Кількість нод змінюється разом із робочим навантаженням – в межах встановленого вами;
- ноди кожного пулу пов'язані між собою **віртуальною мережею** та можуть мати доступ через NFS до даних у віртуальному сховищі (місце надається згідно замовленої квоти);
- деякі з нод кластеру можуть мати публічну **IP-адресу** для доступу ззовні, через Інтернет.

Завдяки цьому, ви можете використовувати переваги функціоналу Kubernetes:

- **оркестрацію** контейнерів – балансування робочого навантаження на систему в рамках проекту;
- автоматизацію запитів на створення/видалення **подів** – сукупностей контейнерів, які пов'язані спільними процесами/алгоритмами та обміном даними;
- запуск/зупинку у **контейнерах** застосунков-мікросервісів, що запаковано разом із потрібними для роботи «залежностями» (бібліотеками та ін.).

## 2. Початок користування послугою з метою тестування

### 2.1. Замовлення тестової конфігурації

Для отримання доступу до послуги, [зареєструйтесь](#) або виконайте вхід до свого облікового запису на порталі [my.gigacloud.ua](#). В хмарі GigaCloud ваш кластер контролюється автоматизованою системою білінгу, що враховує **реальний час** споживання обчислювальних ресурсів, кількість та параметри **працюючих нод** і генерує рахунки на оплату.

Замовляйте Kube-X за стандартною процедурою (тобто, **у режимі самообслуговування через сайт**). Інформацію щодо порядку **конфігурації послуги** наведено нижче.

Спочатку, вам необхідно обрати параметри для вузлів пулу.



**Увага! Конфігурація мастер-ноди створюється автоматично.** Всі робочі ноди пулу створюються **однотипними за запитом**.

Наступним кроком, вам потрібно визначити:

- максимальну кількість вузлів у кластері, до якої може зростати розмір пулу, відповідно до завантаження робочих нод під час роботи;
- розмір постійного сховища даних для вашого кластера;
- кількість «білих» IP-адрес кластеру для доступу до нод “ззовні” – через Інтернет.

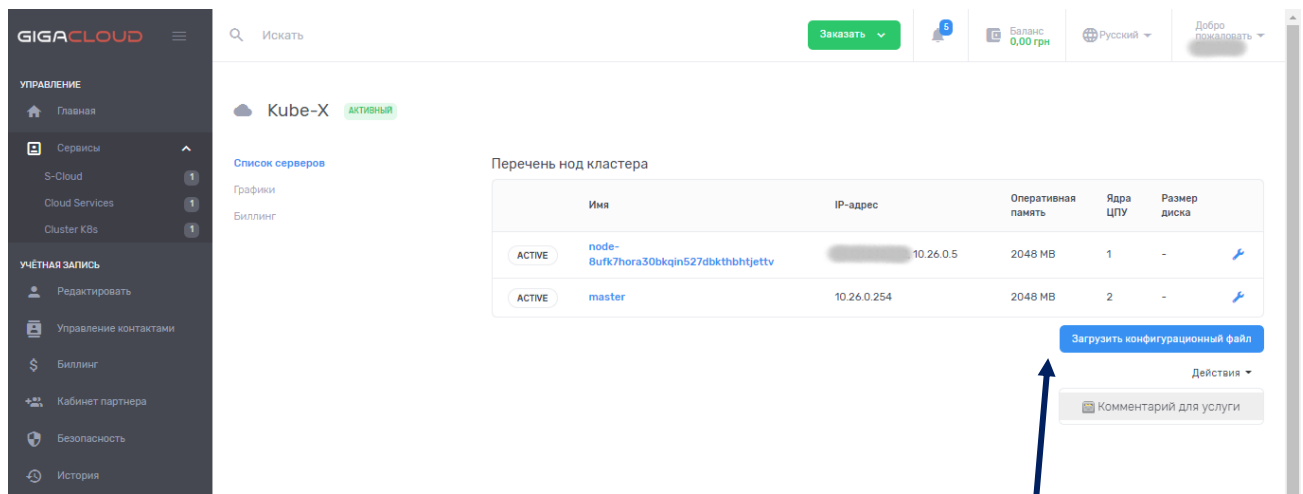
Для завершення замовлення послуги, вам необхідно заповнити відповідні поля форми з контактними даними та натиснути зелену кнопку «**Замовити**» екранного меню:

Після відправки замовлення, ви отримаєте екранне та e-mail повідомлення:



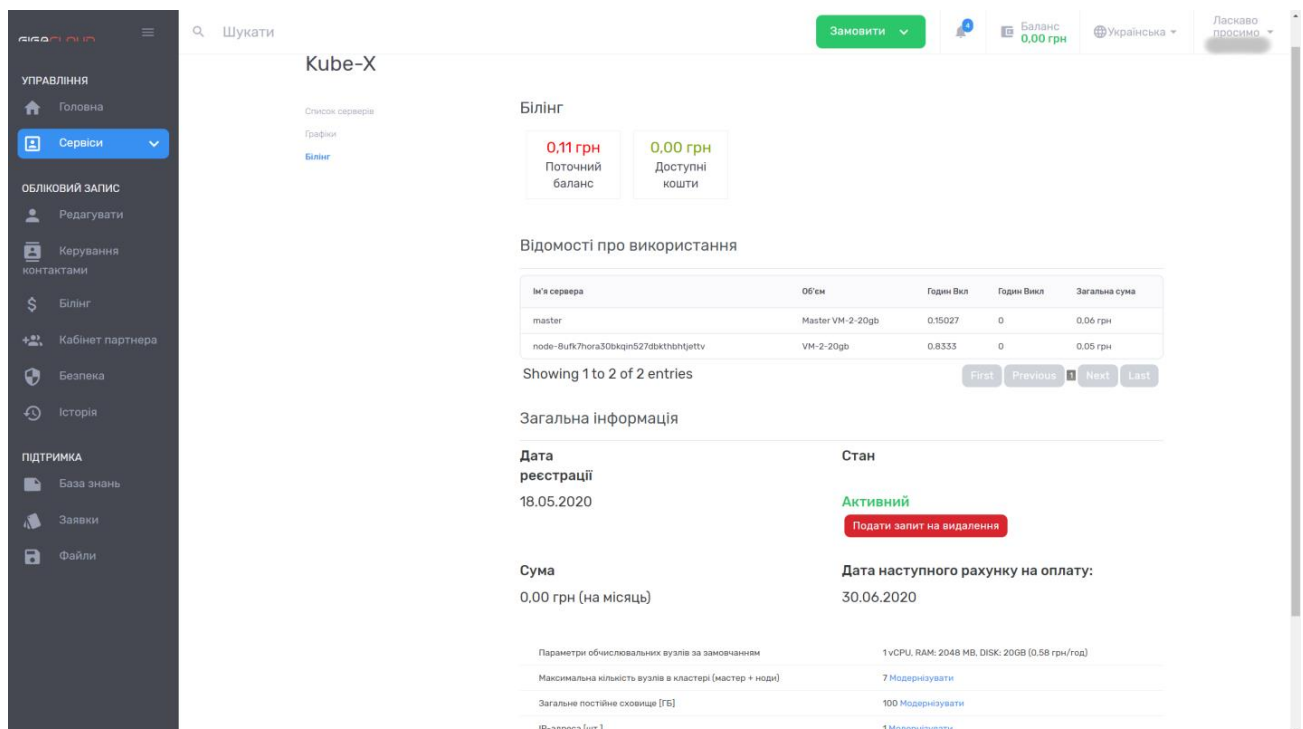
**Увага!** Для створення працездатного кластеру та його конфігурування знадобиться ще декілька хвилин (див. послідовність зображень нижче):

Створення нод починається з мастер-ноди. Після завершення фази «Build» для кожної з нод, їх статус зміниться на «Active».



Ви зможете завантажити свій **конфігураційний файл conf\_k8s.yaml** для користування консольною утилітою **Kubectl**. Приклади використання конфігураційного файлу для налаштувань доступу наведено нижче, у п. 2.2.

Розділ «Білінг» дозволяє контролювати параметри ресурсів та стан оплати послуги:



Для створення додаткового пулу нод з іншими параметрами віртуальних машин (процесори, пам'ять) або декількох пулів у межах вашої квоти ресурсів, використовуйте команди **консолі Kubectl** (див. п.2.2. нижче).

## 2.2. Доступ до ресурсів кластеру (клієнтська частина)

Для початку роботи з кластером **Kubernetes**, вам необхідно **встановити клієнтську частину** – консольну утиліту **kubectl**. Виконайте послідовність команд, залежно від встановленої на вашій робочій станції ОС (див. нижче).

Ubuntu, Debian

```
# sudo apt-get update && sudo apt-get install -y apt-transport-https
# curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key add -
# echo "deb https://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee -a /etc/apt/sources.list.d/kubernetes.list
# sudo apt-get update
# sudo apt-get install -y kubectl
# mkdir ~/.kube
# cp conf_k8s.yaml ~/.kube/config
# kubectl get nodes -o wide
```

Centos, RHEL

```
# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=0
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
# yum install -y kubectl
# mkdir ~/.kube
# cp conf_k8s.yaml ~/.kube/config
# kubectl get nodes -o wide
```

macOS

```
# mkdir ~/.kube/
# cp conf_k8s.yaml ~/.kube/config
# curl -LO https://storage.googleapis.com/kubernetes-release/release/v1.16.0/bin/darwin/amd64/kubectl
# chmod +x ./kubectl
# ./kubectl get nodes -o wide
```

Windows

Завантажте [клієнтську частину](#) застосунку **kubectl.exe** та після його запуску у вікні командного рядку, **виконайте команду**:

```
kubectl --kubeconfig conf_k8s.yaml get nodes
```

Приклад відображення інформації щодо стану нод у пулах кластеру – наведено нижче:

```
C:\Users\user>kubectl --kubeconfig conf_k8s.yaml get nodes
NAME                 STATUS    ROLES    AGE     VERSION
dev-0.novalocal      Ready    <none>   7d17h   v1.14.0
dev-1.novalocal      Ready    <none>   7d17h   v1.14.0
master.novalocal     Ready    master   8d      v1.14.0
node-0.novalocal     Ready    <none>   8d      v1.14.0
node1-0.novalocal    Ready    <none>   8d      v1.14.0
node1-1.novalocal    Ready    <none>   8d      v1.14.0
node1-2.novalocal    Ready    <none>   8d      v1.14.0
node1-3.novalocal    Ready    <none>   8d      v1.14.0
```

### 2.3. Управління інфраструктурою

Щойно створений для вас кластер містить дві ноди:

- мастер (**master.novalocal**);
- перша робоча нода з пула (**node > node-0.novalocal**, назва ноди наведена для прикладу).

У процесі роботи, якщо в межах пулу буде збільшуватися завантаження процесора або оперативної пам'яті (вище, ніж на 70 процентів), у ньому буде створено **додаткову робочу ноду**. Якщо завантаження зменшиться (нижче 30 відсотків), тоді **зайву робочу ноду буде автоматично видалено** системою віртуалізації.

Ручне управління кластером відбувається за рахунок анотацій (команд), наприклад для створення нового пулу:

```
# kubectl annotate nodes master.novalocal pool/<pool_name>=<flavor>.<min>.<max> --overwrite
```

У полях, позначених **<.....>**, вкажіть реальні значення для вашої конфігурації:

- **<pool\_name>** – ім'я пулу (назва першого пулу "by default" – **node**);
- **<flavor>** – тип ноди, параметри (процесор, пам'ять) [з переліку наявних](#), наприклад:  
res\_2048\_p1\_20gb  
res\_4096\_p2\_20gb  
res\_8192\_p3\_20gb  
res\_12288\_p4\_20gb;
- **<min>** – мінімальна можлива кількість нод у даному пулі;
- **<max>** – максимальна припустима кількість нод в даному пулі (якщо дорівнює мінімальній, то кількість нод не буде автоматично змінюватися в залежності від робочого навантаження).



**Увага!** Кількість всіх нод у кластері не повинна бути більшою ніж значення параметру «Максимальна кількість інстансів», який визначається при замовленні послуги та може бути змінений під час користування нею.

**Приклади типових анотацій:**

1. Для створення нового пулу **mysql** розміром від 2 до 10 нод (з ресурсами кожної ноди **res\_8192\_p3\_20gb (3 vCPU, RAM: 8192 MB, DISK: 20GB)**) використовуйте команду:

```
# kubectl annotate nodes master.novalocal pool/mysql=res_8192_p3_20gb.2.10 --overwrite
```

2. Для видалення пула потрібно ввести команду:

```
# kubectl annotate nodes master.novalocal pool/mysql=res_8192_p3_20gb.0 --overwrite
```

3. Для перегляду конфігурації вузлів у пулі, користуйтеся командою

```
# kubectl get nodes -o jsonpath="{.items[0].metadata.annotations.pool/node}"
```

## 2.4. Приклад створення робочої інфраструктури

Одним із прикладів є **розгортання сайту на Wordpress** із базою даних **MySQL**.

В тестовому кластері створимо три пули:

1. Пул **“front”** – для «фронтенд», доступу до веб-сайту через проксі-сервер.



**Увага!** До ноди **front-0** даного пулу потрібно приєднати публічний IP для доступу ззовні, через Інтернет.

Створіть пул **“front”** з однієї ноди:

```
# kubectl annotate nodes master.novalocal pool/front=res_2048_p1_20gb.1.1 --overwrite
```



**Увага!** Процес створення ноди може зайняти декілька хвилин.

Після створення серверу **«front-0»**, через меню **«Інтерфейси»** додайте **«Floating IP-адресу»**, як наведено у прикладі на зображенні нижче:

MAC-адреса	IP-адреса	Стан
FA:16:3E:68:50:6D	10.26.0.5	ACTIVE

Занотуйте отриману **публічну Floating IP-адресу** для подальших налаштувань вашої інфраструктури.

2. Пул **“node”** – для розгортання веб серверу **WordPress**:

```
# kubectl annotate nodes master.novalocal pool/node=res_8192_p3_20gb.1.1 --overwrite
```

3. Пул **“mysql”** – для створення бази даних **MySQL**:

```
# kubectl annotate nodes master.novalocal pool/mysql=res_8192_p3_20gb.1.1 --overwrite
```

В результаті, ноди будуть створені (загалом, **це може зайняти до 10 хвилин**).

Виконайте команду для перевірки їх параметрів:

```
# kubectl get nodes -o wide
```

Отримайте відповідь, подібну до наведеної нижче:

NAME	STATUS	ROLES	AGE	VERSION	INT-IP	EXT-IP	OS-IMAGE	KERNEL-VERSION	CONTAINER-RUNTIME
front-0.novalocal	Ready	<none>	21h	v1.14.0	10.3.0.16	<none>	CentOS Linux 7 (Core)	3.10.0-957.10.1.el7.x86_64	docker://1.13.1
master.novalocal	Ready	master	25h	v1.14.0	10.3.0.254	<none>	CentOS Linux 7 (Core)	3.10.0-957.10.1.el7.x86_64	docker://1.13.1
mysql-0.novalocal	Ready	<none>	25h	v1.14.0	10.3.0.13	<none>	CentOS Linux 7 (Core)	3.10.0-957.10.1.el7.x86_64	docker://1.13.1
node-0.novalocal	Ready	<none>	25h	v1.14.0	10.3.0.4	<none>	CentOS Linux 7 (Core)	3.10.0-957.10.1.el7.x86_64	docker://1.13.1

Створіть тестовий сертифікат:

```
openssl req -new -newkey rsa:4096 -x509 -sha256 -days 365 -nodes -out MyCertificate.crt -keyout MyKey.key
```



Для розгортання сайту на WordPress, завантажте файли через браузер:

<https://gl.gigacloud.ua/gigacloud/caas-public/tree/master/WordPress-Samples>

або користуючись командами Linux:

wget <https://gl.gigacloud.ua/gigacloud/caas-public/-/archive/master/caas-public-master.zip> unzip caas-public-master.zip

Перейдіть до каталогу з файлами `cd caas-public-master/WordPress-Samples` та відредагуйте завантажені демо-файли, з урахуванням ваших параметрів:

### 1. nano add-namespace.yaml

apiVersion: v1

kind: Namespace

metadata:

name: Назва\_простору імен



**Увага!** Назва простору імен (Namespace) має відповідати назві сайту, але:

- не повинна містити спецсимволи;

- замість крапок потрібно писати дефіси, наприклад `wp-gigacloud-ua`

### 2. nano wordpress-cli.yaml

apiVersion: v1

kind: Secret

metadata:

name: wp-tls

data:

tls.crt: «тіло» сертифікату (без BEGIN та END CERTIFICATE) в один рядок, перед «хеш» пробіл

tls.key: «тіло» ключа (без BEGIN та END KEY) в один рядок, перед «хеш» пробіл



**Увага!** Для перетворення хешу сертифіката до формату одного рядку, користуйтеся командою: `cat MyCertificate.crt | tr -d '\r\n' > MyCertificate.txt` (аналогічно, для файлу `MyKey.key`)

type: kubernetes.io/tls

---

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

name: example-ingress

annotations:

ingress.kubernetes.io/rewrite-target: /

nginx.ingress.kubernetes.io/ssl-redirect: "false"

spec:

tls:

- hosts:

- wp.gigacloud.ua

secretName: wp-tls

rules:

- host: Ваш.домен.com

http:

paths:

- path: /

backend:

serviceName: wordpress

servicePort: 80

### 3. nano mandatory.yaml

Дізнайтесь, яку **приватну IP-адресу** внутрішній мережі отримала нода пулу **front**

Виконайте команду: `kubectl get nodes -o wide`

Отримайте назву ноди (наприклад, «**front-0.novalocal**») та застосуйте у наступній команді:

`kubectl get nodes front-0.novalocal -o wide` і отримайте відповідь:

```
front-0.novalocal Ready <none> 22h v1.14.0 10.3.0.16 <none> CentOS Linux 7 (Core) 3.10.0-957.10.1.el7.x86_64 docker://1.13.1
```

Відповідно, у файлі відредагуйте рядок з IP-адресою (наприкінці файлу):

```
externalIPs:
  - 10.3.0.16
```

---

Активуйте інфраструктуру:

`kubectl create -f mandatory.yaml -f add-namespace.yaml`

`kubectl --namespace <Назва_простору_імен> create -f wordpress-deploy.yaml -f wordpress-cli.yaml`

<Ваш.домен.com> має посилатися на публічну IP-адресу, яку приєднано до «фронтенд» (див. п.2.4. Стор.7). **Виконайте реєстрацію на сервері доменних назв DNS**. Створіть запис з назвою вашого домену та публічною IP адресою, що пов'язано до «фронтенд».

Виконайте у консолі команду `nslookup` **Ваш.домен.com**

Відповідь має містити публічну IP адресу, що пов'язано до «фронтенду».

Після реєстрації, зачекайте 5-10 хвилин для оновлення DNS-записів.

Перевірте доступність сайту **Ваш.домен.com**

Имя	TTL	Тип	Детали
mail	14400	A	белый IP адрес, привязанный к фронтэнду
@	14400	A	
www	14400	A	

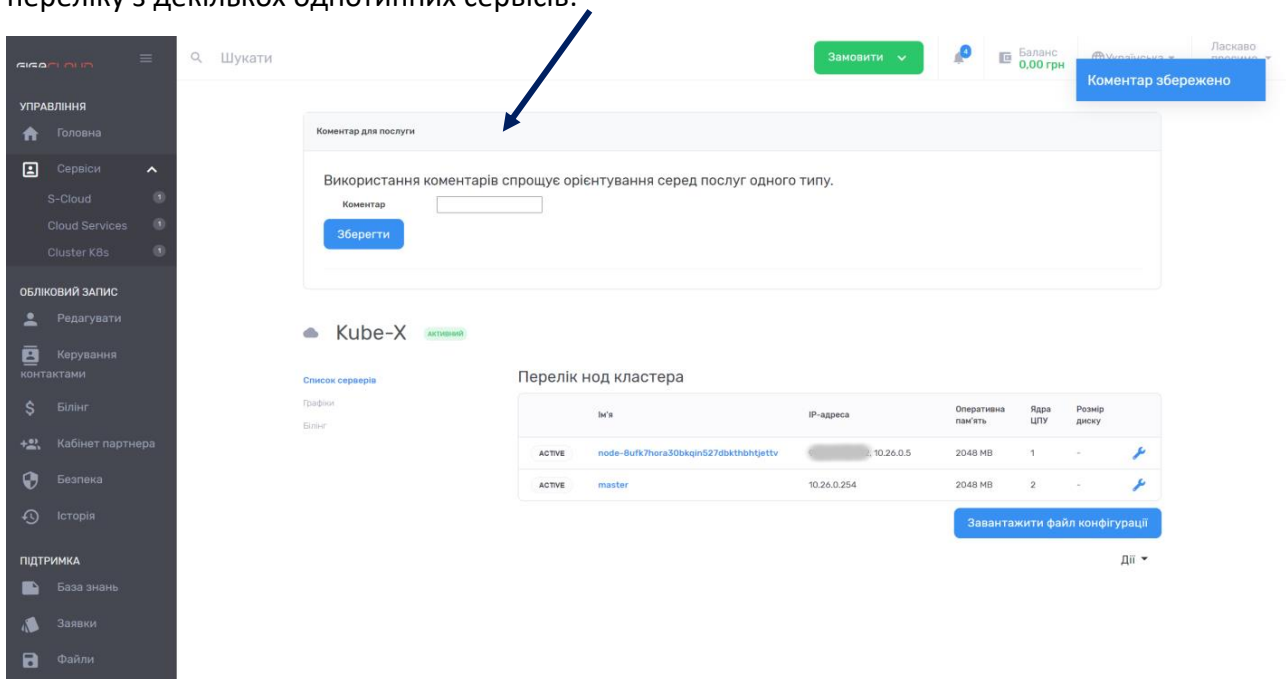
Виконайте інсталяцію **WordPress**

**та перевірте** працездатність тестового проекту (див. зображення нижче):

### 3. Поширені запитання і відповіді

Як додати новий пул до вже існуючого кластеру?

Користуючись командами-анотаціями **KubectI**, ви маєте можливість створювати додаткові пули з робочими серверами (нодами), у складі працюючого кластеру (див. п. 2.3.) та контролювати стан кластеру, користуючись вашим клієнтським порталом (кабінетом користувача). Для зручності, ви можете додавати свої коментарі, що спрощує навігацію у переліку з декількох однотипних сервісів:



Додатковий функціонал підключення «білих» (публічних) IP адрес

Для доступу до ноди (серверу) ззовні, через Інтернет, необхідно приєднати до неї так звану «білу» (публічну) IP адресу.

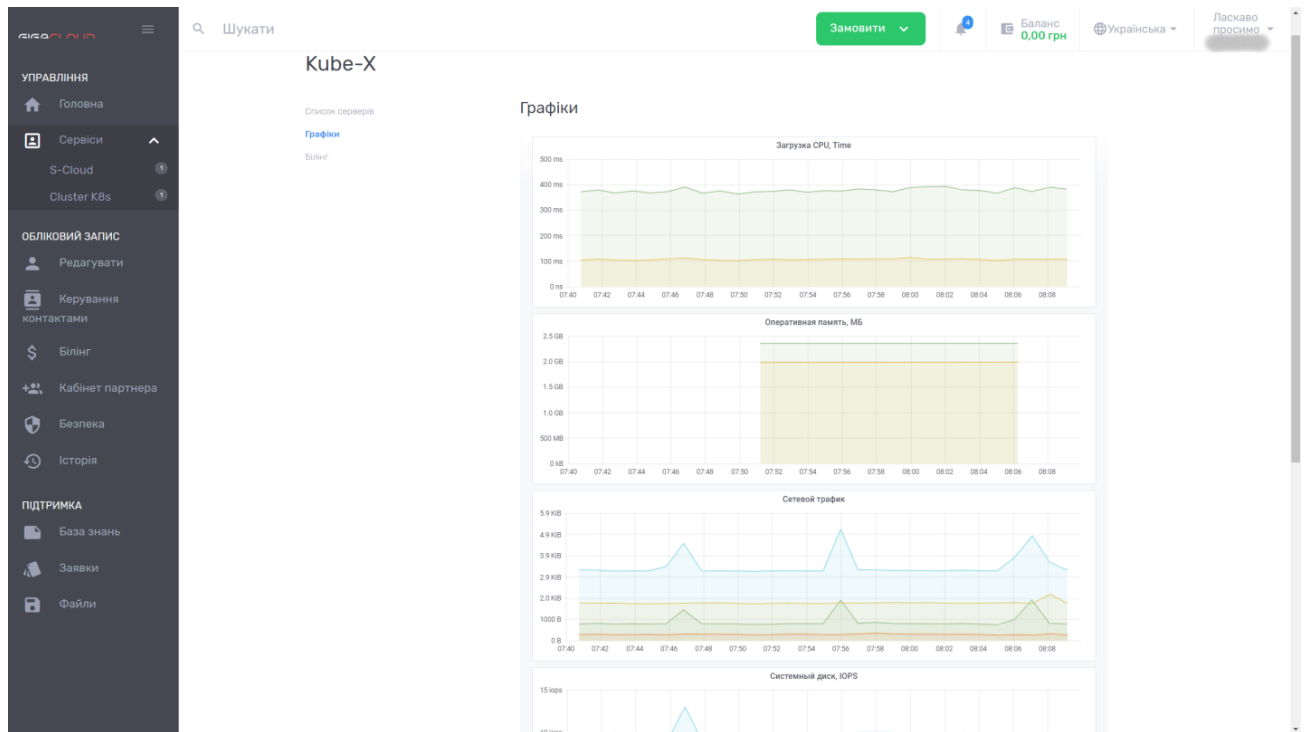
Для цього, користуйтеся **анотацією** KubectI:

**kubectI annotate nodes front.novalocal ip/front=1 --overwrite**

Занотуйте отриману публічну IP-адресу для подальших налаштувань вашої інфраструктури.

Як дізнатися про робочі навантаження нод кластеру?

Користуйтеся розділом «Графіки» в екранному меню послуги:



Як деактивувати тестову інфраструктуру та видалити непотрібні ноди кластеру?

Для деактивування створеної інфраструктури (див. 2.4.), виконайте послідовність команд Kubectl:

```
kubectl --namespace <Назва_вашого_сайта> delete -f ./wordpress-cli.yaml
kubectl --namespace <Назва_вашого_сайта> delete -f ./wordpress-deploy.yaml
kubectl delete -f ./add-namespace.yaml
kubectl delete -f ./mandatory.yaml
```

Користуючись командами - анотаціями **Kubectl**, ви маєте можливість видаляти пули з нодами зі складу працюючого кластеру. Для видалення ноди, виконайте послідовність **Kubectl**:

```
kubectl annotate nodes master.novalocal pool/front=res_2048_p1_20gb.0 --overwrite
kubectl annotate nodes master.novalocal pool/mysql=res_8192_p3_20gb.0 --overwrite
```

Виникли питання – звертайтеся до нашої [технічної підтримки!](#)

У полі “**Тема**” екранної форми вкажіть: «Послуга **Kube-X**».