CloudTide Project

Overview

In enterprises that have private clouds, their capacity is usually designed to meet the average or peak time usage. When usage drops, the resources are very likely to keep running but stay idle. Although those resources are running idle, they are still consuming power, cooling, space and operational labor to maintain them. These costs are wasted. Considering the huge number of companies (with private cloud) in the world, the idle resources and the waste are huge.

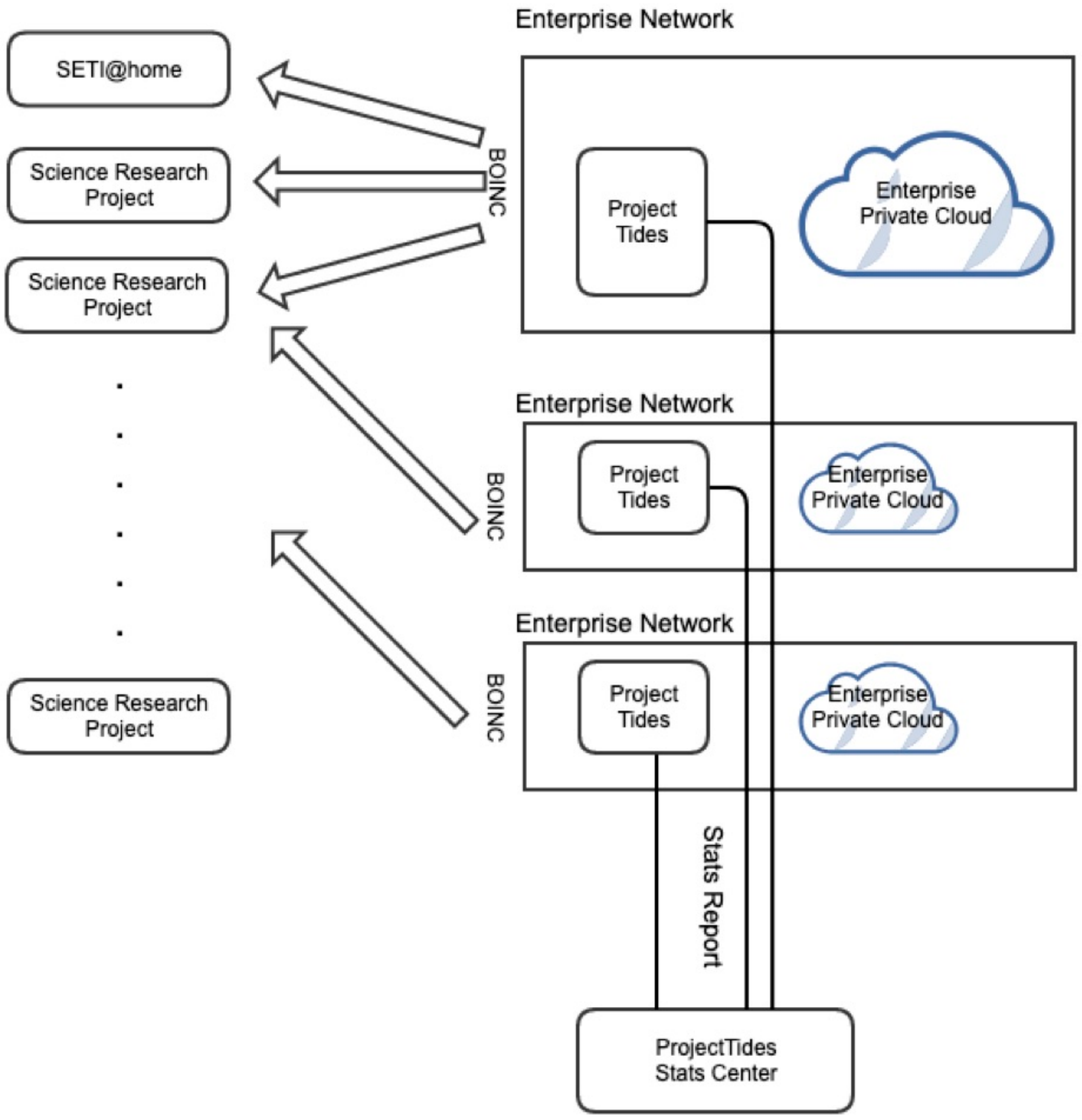
On the other side, there are many organizations in the world that need computing resources, for example, many non-profitable organizations (NPO) and science research organizations. They may not have enough budget to purchase all of the required capacity or the procurement and installation may take a long time to complete. Therefore, they have a strong need to leverage the existing resources in the world. I would like to propose this project to bridge the requirements of such organizations with the idle resources in the enterprises.

Although we don't have the market data of how large the requirements from such organizations are, there are existing solutions in this area to meet the needs. For example, the famous SETI@home project run by UC Berkeley is a good example to unveil such cases. SETI@home calls people to donate their idle personal devices, from home lab to desktop/laptop to mobile devices. This project has been running for more than 20 years and during peak time, there were more than 1 million devices contributing computing resources to the project. Such kind of computing is also known as volunteer computing. However, SETI@home only includes personal devices. They have collaborated with companies like IBM to include employees' devices to run SETI@home clients, but there is no company contributing private cloud resources to SETI@home.

In addition, SETI@home is based on the BOINC program to distribute the jobs to the individual devices. And there are about 30 science projects using BOINC today. Our project wants to support BOINC and thus is able to contribute idle private cloud resources to the various of cutting-edge science research projects in the world.

Organizations/programs like SETI@home are the demanders of our project, while enterprises with private cloud are the suppliers. As VMware's product is the most used virtualization software in the market, our project assumes that the suppliers have vSphere in their private cloud and thus we can use vSphere API to communicate with the private cloud.

From a single enterprise viewpoint, they can contribute idle resources to multiple science research projects and from a science research project viewpoint, they can distribute computing jobs to multiple enterprises. We want to have a stats center to collect usage data.



The Architecture

Project Tides is the middleman between the demander and the supplier. It will have the following modules.

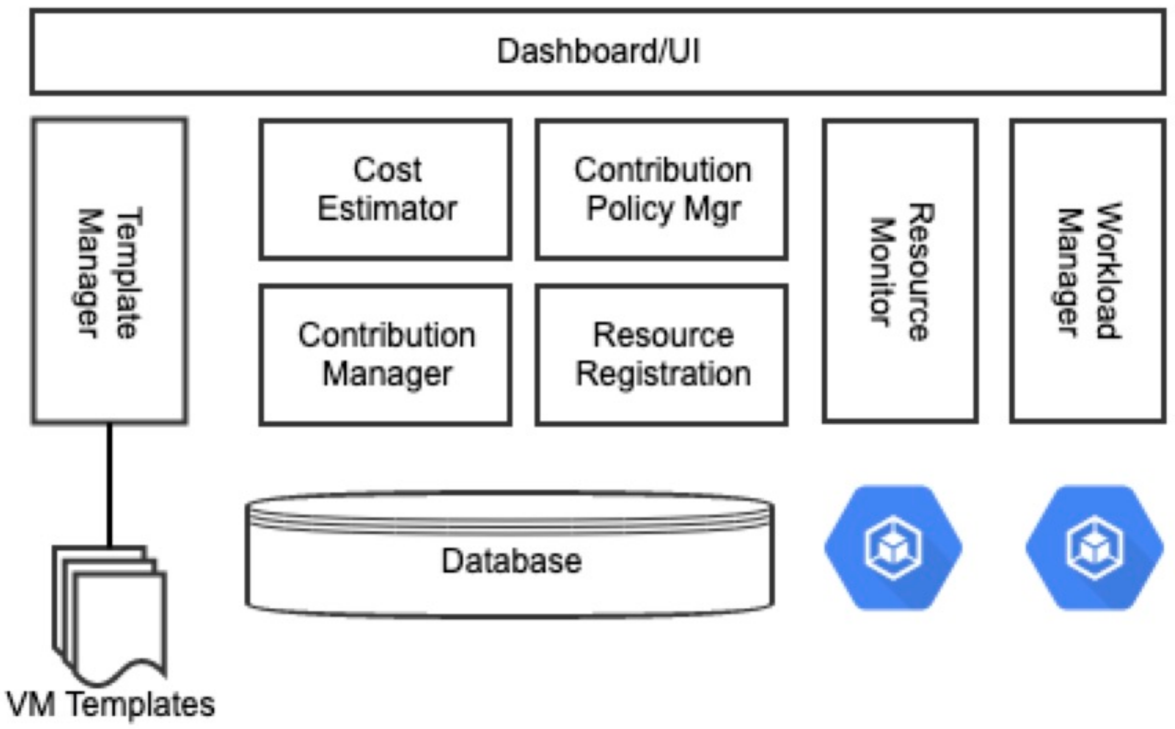
- Resource Registration
  This module will provide an interface to register or deregister idle resources. It will allow the administrators to add/remove/update/enable/disable private cloud resources in this platform. It now should support vSphere (vCenter/ESXi/VCD) as the cloud provider and use vSphere API or vMomi API to connect to the infrastructure.

- Resource Monitor
  Once a resource is registered in the platform, we need to continuously monitor the usage on the target resource. vSphere/vMomi API provides interface to query the real-time usage data and we need to poll the data and pass them to a queue. Then other module can kick off the corresponding work based on the defined policies. This module should be able to allow users to choose whether to run in a K8S cluster or in some static worker VMs, depending on the number of workloads.

- Contribution Policy Manager
  This module will define the 'idle' state for the registered cloud infrastructure. For example, CPU usage below 30% can be defined as idle and trigger the contribution workloads provisioning. It should also define when to stop the new workload deployments and when to destroy the workloads. Different resources may have different thresholds.

- Workload Manager
  This module will execute the deployment and destroy job. It will monitor the usage data in the queue and compare it with the contribution policy. Once a policy is satisfied, it will execute the defined operation. This module should be able to allow users to choose whether to run in a K8S cluster or in some static worker VMs, depending on the number of workloads.

- Template Manager
  This module controls the VM template files in the platform. All of the computing jobs need to run in VMs and the VM should be deployed from those templates. In the template, the administrators can install necessary software and patches to compliant with the enterprise security policy /guideline.
- Contribution Manager
  In this module, the administrator can define which external project/site the enterprise wants to contribute the idle resources to.
- Cost Estimator
  This module will try to estimate the value the enterprise has contributed to the research projects.
- Dashboard
  The dashboard will provide UI for the above modules, show the contribution history and the current status of the workloads.

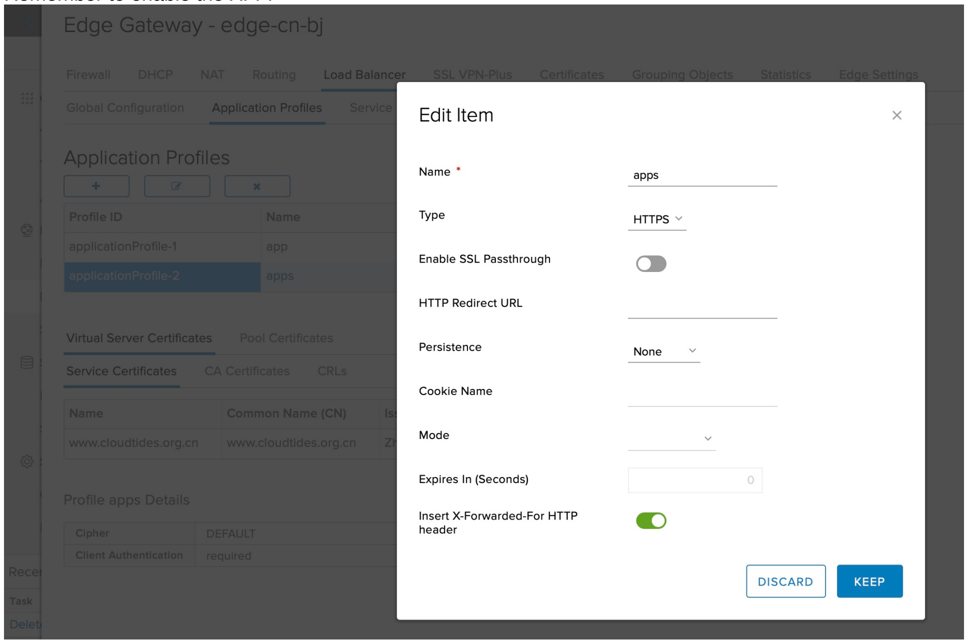


For more introductions about this project, you can refer to https://github.com/cloudtides/CloudTides/wiki/CloudTides----Elastic-Platform-on-Idle-Cloud-Resources

How to expose the service of a specific VM through Load Balancer

One important feature of CloudTides is that it can automatically expose the service of a specific VM through Load Balancer, this feature is realized in function ***ExposePorts*** at https://github.com/cloudtides/CloudTides/blob/master/tides-server/pkg/handler/vapp_api.go. And there are several steps to leading to the achievement of this feature. Suppose we already have an Edge Gateway and a valid public IP address in the given datacenter.

1. Create an Application Profile, commonly you can just set its type as "HTTP" (for "HTTPS" I will talk about this in the two-way SSL certificate part). Remember to enable the XFF.



Application Profiles

Profile ID	Name
applicationProfile-1	app
applicationProfile-2	apps

Virtual Server Certificates · Pool Certificates · Service Certificates · CA Certificates · CRLs

Name	Common Name (CN)
www.cloudtides.org.cn	www.cloudtides.org.cn

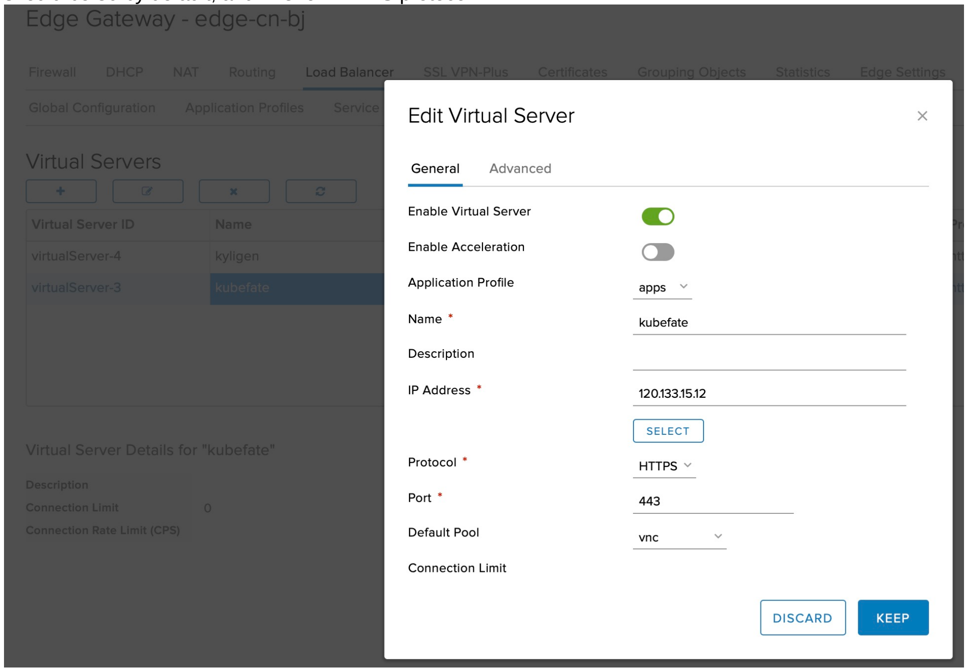
Profile apps Details

Cipher	DEFAULT
Client Authentication	required

Edit Item

Name *	apps
Type	HTTPS
Enable SSL Passthrough	off
HTTP Redirect URL	
Persistence	None
Cookie Name	
Mode	
Expires In (Seconds)	0
Insert X-Forwarded-For HTTP header	on

2. Set up a Virtual Server. Choose the Application Profile you created at the last step for this Virtual Server, enter the public IP address you have, and remember that the protocol of your Virtual Server should match with your Application Profile. For the port, if you are using HTTP protocol, it should be 80 by default, and 443 for HTTPS protocol.



Virtual Servers

Virtual Server ID	Name
virtualServer-4	kyligen
virtualServer-3	kubefate

Virtual Server Details for "kubefate"

Description	
Connection Limit	0
Connection Rate Limit (CPS)	

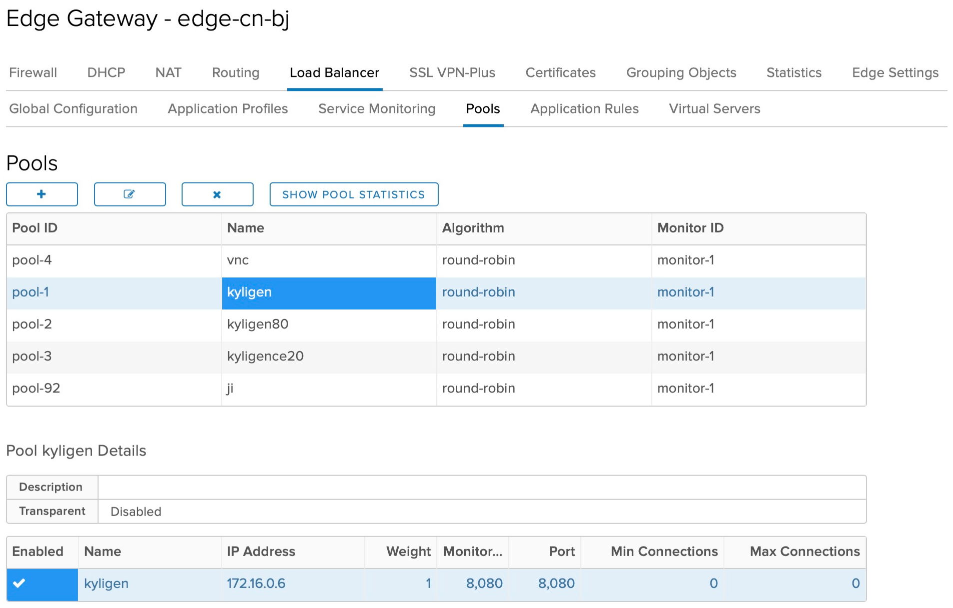
Edit Virtual Server

General · Advanced

Enable Virtual Server	on
Enable Acceleration	off
Application Profile	apps
Name *	kubefate
Description	
IP Address *	120.133.15.12
Protocol *	HTTPS
Port *	443
Default Pool	vnc
Connection Limit	

3. You should have a Service Monitor of TCP type. If don't, create one by yourself.
4. Create LoadBalancer Pools for the specific VM. One pool maps to one port of the VM. For each LoadBalancer Pool, choose its monitor as the one you created or have at last step, then insert a member in the pool, the IP Address should be the private IP address of the VM, the Port and

Monitor Port should be the port you want to expose. In the figure below, for example, we want to expose the 8080 port for the VM with the internal IP address 172.16.0.6.



Pool ID	Name	Algorithm	Monitor ID
pool-4	vnc	round-robin	monitor-1
pool-1	kyligen	round-robin	monitor-1
pool-2	kyligen80	round-robin	monitor-1
pool-3	kyligence20	round-robin	monitor-1
pool-92	ji	round-robin	monitor-1

Pool kyligen Details

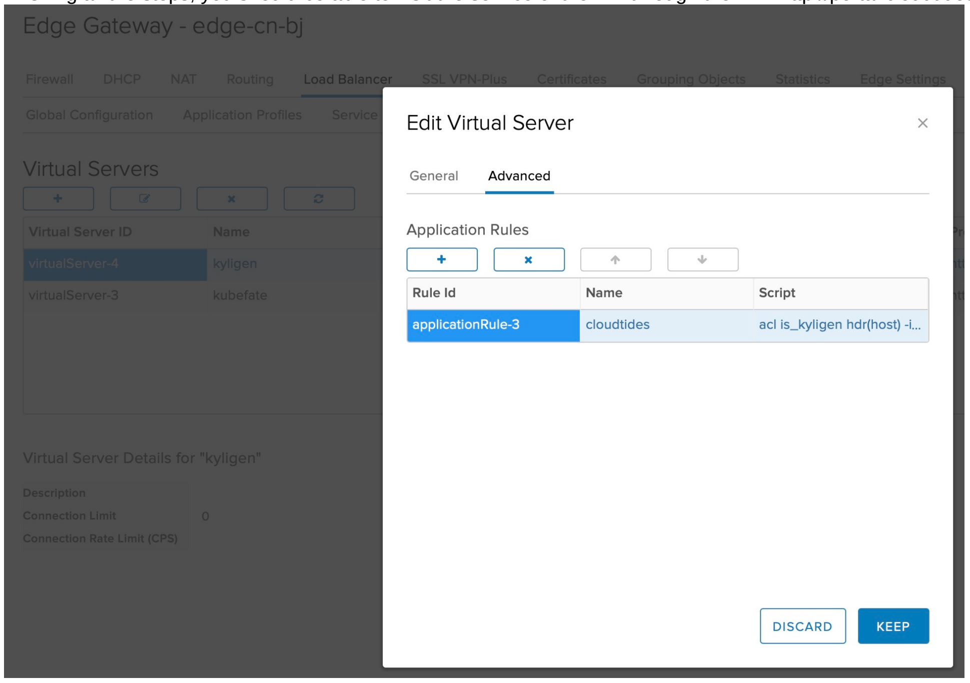
Description	
Transparent	Disabled

Enabled	Name	IP Address	Weight	Monitor...	Port	Min Connections	Max Connections
✓	kyligen	172.16.0.6	1	8,080	8,080	0	0

5. Create Application Rule for your service exposing. To finish this step, you should make sure that you have a valid domain name (in this case, it is "cloudtides.vthink.cloud"), then you can edit the script of the Application Rule according the following example:
   acl is_portal hdr(host) -i portal.cloudtides.vthink.cloud

   use_backend portal if is_portal

   First you define a boolean value as "*is_portal*", and it will return true if the URL that user tried to reach out is "*portal.cloudtides.vthink.cloud*". And then you define that if "*is_portal*" is true, LoadBalancer will redirect user to the backend "*portal*". Here "*portal*" should be replaced by the name of LoadBalancer Pool you created before.
6. At last, go back to the Virtual Server at the second step, edit it, go to the Advanced Option, add the Application Rules at the last step. After finishing all the steps, you should be able to visit the service of the VM through the link "http://portal.cloudtides.vthink.cloud"

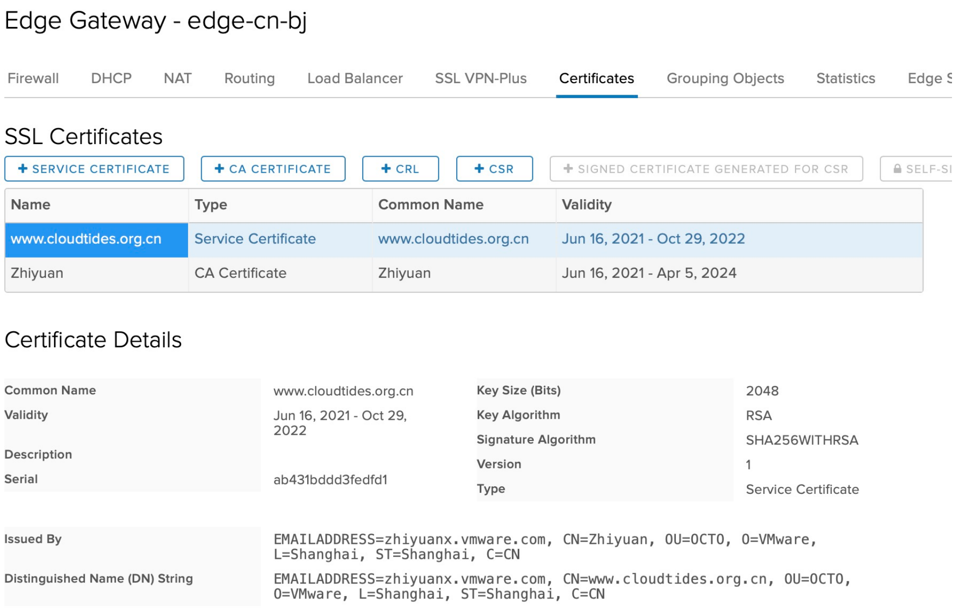


Rule Id	Name	Script
applicationRule-3	cloudtides	acl is_kyligen hdr(host) -i...

How to enable the two-way SSL certification for a exposed service

Sometimes the user may not want others to visit the service exposed on the CloudTides, then we have to enable the two-way SSL certification for that exposed service.

1. Prepare a root CA certificate and a service certificate (for user), and then sign the service certificate with root CA certificate. For more details about this step you can refer to https://gist.github.com/fntlnz/cf14feb5a46b2eda428e000157447309
2. Upload the CA certificate and service certificate to the Edge Gateway.



Edge Gateway - edge-cn-bj

SSL Certificates

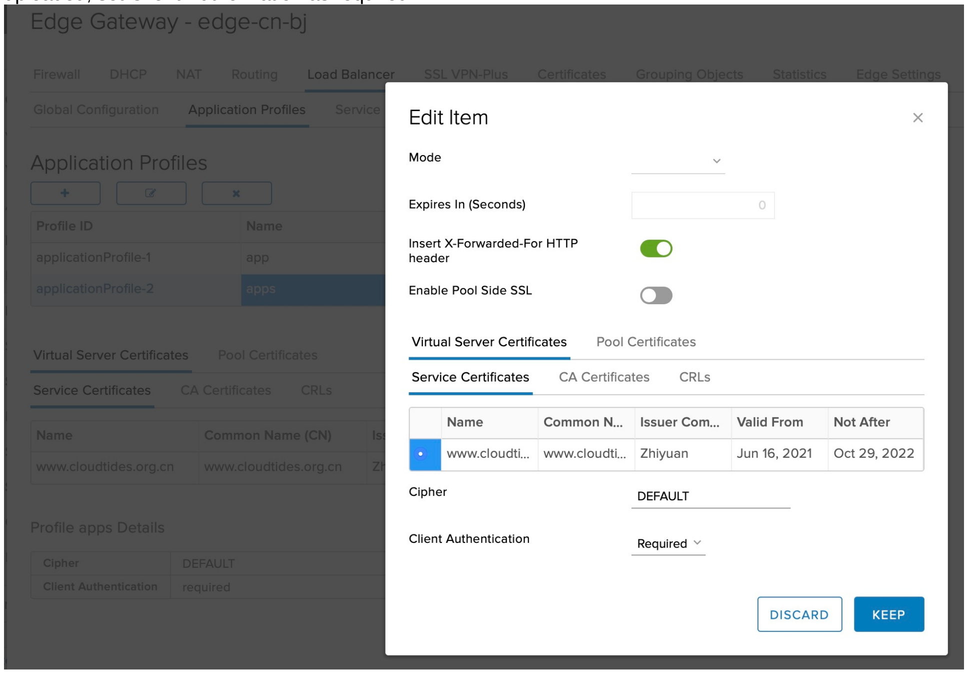
Name	Type	Common Name	Validity
www.cloudtides.org.cn	Service Certificate	www.cloudtides.org.cn	Jun 16, 2021 - Oct 29, 2022
Zhiyuan	CA Certificate	Zhiyuan	Jun 16, 2021 - Apr 5, 2024

Certificate Details

Common Name	www.cloudtides.org.cn	Key Size (Bits)	2048
Validity	Jun 16, 2021 - Oct 29, 2022	Key Algorithm	RSA
		Signature Algorithm	SHA256WITHRSA
Description		Version	1
Serial	ab431bddd3fedfd1	Type	Service Certificate

Issued By	EMAILADDRESS=zhiyuanx.vmware.com, CN=Zhiyuan, OU=OCTO, O=VMware, L=Shanghai, ST=Shanghai, C=CN
Distinguished Name (DN) String	EMAILADDRESS=zhiyuanx.vmware.com, CN=www.cloudtides.org.cn, OU=OCTO, O=VMware, L=Shanghai, ST=Shanghai, C=CN

3. Create a new Application Profile in Load Balancer with type "HTTPS", enable the XFF, select the CA certificate and service certificate you uploaded, set Client Authorization as required.



Profile ID	Name
applicationProfile-1	app
applicationProfile-2	apps

Cipher: DEFAULT
Client Authentication: required

Edit Item

Mode:
Expires In (Seconds): 0
Insert X-Forwarded-For HTTP header: on
Enable Pool Side SSL: off

	Name	Common N...	Issuer Com...	Valid From	Not After
●	www.cloudti...	www.cloudti...	Zhiyuan	Jun 16, 2021	Oct 29, 2022

Cipher: DEFAULT
Client Authentication: Required

4. Create a Virtual Server with protocol as "HTTPS".
5. Re-do the steps 3-6 at "How to expose the service of a specific VM through Load Balancer".
6. Now you can visit the service with HTTPS, and remember to upload the service certificate on your web browser.

Though the method above can help you to enable the two-way SSL certification for a exposed service, but there still exist some potential problems that needed to be handled in the future:

1. The CA certification in an Application Profile is binding to the virtual server, and a virtual server can only apply one Application Profile, this means if we have two or more services need to be exposed with different certification, then we must set up different virtual servers, and each virtual server should have a unique pair of IP and port. Currently we only have two virtual server with the same IP but different port.
2. The automation of this part in CloudTides is hardcoded in the function ***ExposePorts*** at https://github.com/cloudtides/CloudTides/blob/master/tides-server/pkg/handler/vapp_api.go, and we just created two Application Profile *cloudtides* and *kubefate*, thus if in the future we need to allow different services to have different certifications, this part of code should be changed accordingly.

How to build a template for docker deployed two-party KubeFATE cluster on CloudTides

KubeFATE is a federal learning platform, and it could involve several parties to build a cluster. To make the template, you first should prepare three virtual machines with docker on them. One is deployment machine and the other two are target nodes (each represents different party). In theory the target nodes can also be the deployment machine, so you can make it with only two virtual machines. For the details about how to build the cluster manually, you can refer to https://github.com/FederatedAI/KubeFATE/tree/master/docker-deploy and https://www.bilibili.com/video/BV1ZE41157CV?from=search&seid=6973199025800232902. Here I'll mainly explain what I did to allow CloudTides to automatically deploy a two-party KubeFATE cluster.

1. Create a vAPP with three VMs, here I recommend to use CentOS7 instead of Ubuntu 18.04 (reason for this will be explain later), each VM should have vmware tools installed. Two of them are regarded as target node and should meet the requirements mentioned in https://github.com/FederatedAI/KubeFATE/tree/master/docker-deploy. Another one is the deployment machine, and it should download the release file from https://github.com/FederatedAI/KubeFATE/releases, unpack and place the directory at root path, rename it to be 'docker-deploy'. (make sure you can find the parties.conf file at the path '/root/docker-deploy/parties.conf').
2. Enable deployment machine to apply Password-less SSH Login to two target nodes. Also you should disable SSH host key checking according to https://www.shellhacks.com/disable-ssh-host-key-checking/#:~:text=By%20default%2C%20the%20SSH%20client,remote%20host%20over%20SSH%20protocol. If not the deployment cannot be done automatically.
3. In the deployment machine you should download CloudTides client from https://github.com/cloudtides/CloudTides/tree/master/tides-server/client. This is a client which will collect information of the other two target nodes and modified the parties.conf file (https://github.com/FederatedAI/KubeFATE/blob/master/docker-deploy/parties.conf) accordingly. Put the client at '/root/client/' path, and also write a bash script called test.sh at the path '/root'.
4. Write guest customization script for the deployment machine. This part is automatically done by CloudTides server, you can refer to the function ***DeployVAPP*** at https://github.com/cloudtides/CloudTides/blob/master/tides-server/pkg/handler/vapp_api.go. The guest customization script will be run when the VM is creating (which means that it is possible when the script is running, the network is not available), and there exist a maximum running time of the script, it will be killed if it cannot be done within about 5 mins. Another problem with the guest customization script is that it cannot be run on a Ubuntu 18.04 OS (for the solution you can refer to https://blog.ntitta.in/?p=344), that's why I recommend you to use CentOS7 at the very beginning.
5. Finally you can just save the whole vAPP as a template in catalog.

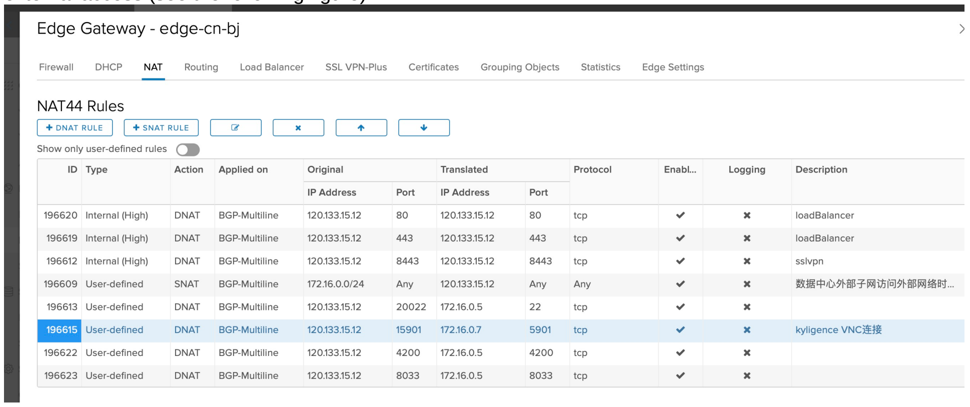
After finishing all these steps, you should be able to run a two-party KubeFATE cluster on CloudTides.

How to build a single node sandbox of Kyligence

Kyligence is a big data platform and we successfully set up a single node sandbox of Kyligence on CloudTides, here are some tips about how to realize this feature, for more details about how to install and run Kyligence, please refer to the official documents: https://docs.kyligence.io/books/v4.3/zh-cn/installation/prerequisite.cn.html

- To install and run the Kyligence, you MUST first have a proper hadoop platform. According to its official documents, the following hadoop platform with specific version number is tested and verified by Kyligence: Cloudera CDH 5.8 / 6.1 / 6.2 / 6.3, Hortonworks HDP 2.4, FusionInsight C70 / 6.5.1, Apache Hadoop 2.7.2 Kyligence Enterprise 4.3.3), MapR 6.1.0 Kyligence Enterprise 4.3.3). Here we choose to use CDH (because CDH is the most frequently used hadoop platform for Kyligence according to their tech staffs). But the problem is that the CDH has no more community version since February 2021, unless you have an official license, you can no longer install CDH from Cloudera official website. One solution for this problem is to get an unofficial community version CDH. In our case, I got a OVA file which contains CDH 5.10 from one staff of Kyligence. Remember that even if you got the community version of CDH, since Cloudera stop the support for free version CDH, all official mirrors of CDH are no longer valid, you cannot update your Cloudera Manager.
- After you successfully installed CDH, or have a VM which had CDH installed, the next step is to download and install Kyligence. Before you run the Kyligence, don't forget to test the validation of your environment, make sure that hdfs, hive works well. For this part, just follow the official documents. If you cannot successfully run Kyligence, you can take the short cut: just to get a OVA file which already has Kyligence installed.
- Another problem I met during the process is that all the OVA file I got from Kyligence, is built on CentOS 6. However, since December 2020, RedHat decided to stop the official support for CentOS 6.X, in this case, all the URL for mirrors used in yum is no longer valid, if you want to use yum to install any package, you may get "*All mirror URLs are not using ftp, http[s] or file*" error, to fix this problem, you can refer to How to fix CentOS 6 error: YumRepo Error: All mirror URLs are not using ftp, http[s] or file - Computer How To for further help. But even if you update the basic yumRepo, you may failed when you typed "*yum update*", because as I have mentioned before, Cloudera also stopped the support for CDH free version, which means all the repo of cloudera is invalid now. In order to let yum works well, you need to remove all the invalid repo of yum. Go to /etc/yum.repos.d/, remove or rename all the .repo file starts with word "*cloudera*".
- After we set up the whole environment for Kyligence, we should write an automation service that can start Kyligence on boot. After several attempt, I choose to use ***chkconfig*** service to achieve this, the service is wrote at "/etc/init.d/kyligen" and for more details about chkconfig you can refer to https://www.thegeekdiary.com/how-to-enable-or-disable-service-on-boot-with-chkconfig/. There are several reasons I choose chkconfig:
  First, systemd is not available under CentOS 6, it requires CentOS 7 or above
  Second, rc.local and customization script in VCD is not working because Kyligence requires the environment from Cloudera, and Cloudera in the given OVA template is started with chkconfig, and if you use rc.local to start Kyligence, you will find at that time Cloudera is not ready, and you will fail. In this case, I have to choose chkconfig and set the start/boot order of service kyligen as low as possible, so that I can make sure when Kyligence is starting, Cloudera is ready to use.
- For CDH, if you want to change the configuration of the hadoop cluster, like change the memory size of each yarn node, currently you can only do this in the Cloudera Manager through GUI (maybe there are some ways that allow you to modify the configuration through command line, but currently I don't know it), but ThinkCloud web console currently have a poor support for the GUI connection, as well as the VMware Remote Console. In this case, in order to enable GUI control of the Cloudera Manager, one possible way is to set up a VNC server in the VM, and then you can use a VNC client which is able to access to the VM, and build up a remote GUI control. Take ThinkCloud for example, after you set up a VNC server in a VM with internal IP address 172.16.0.7 and expose to the port 5901, you can create a NAT44 Rule in the edge-gateway to allow

external access (see the following figure).



Edge Gateway - edge-cn-bj

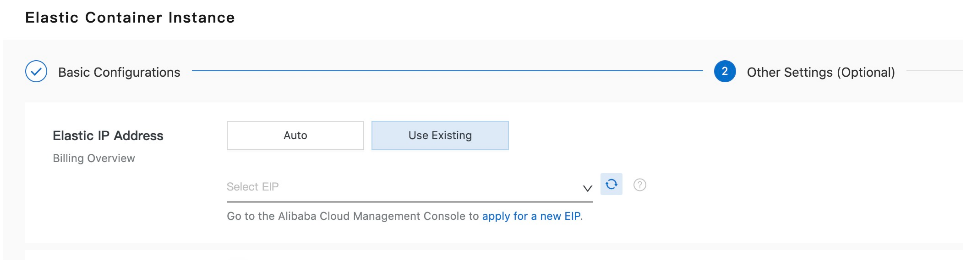
NAT44 Rules

ID	Type	Action	Applied on	Original		Translated		Protocol	Enabl...	Logging	Description
				IP Address	Port	IP Address	Port				
196620	Internal (High)	DNAT	BGP-Multiline	120.133.15.12	80	120.133.15.12	80	tcp	✓	✗	loadBalancer
196619	Internal (High)	DNAT	BGP-Multiline	120.133.15.12	443	120.133.15.12	443	tcp	✓	✗	loadBalancer
196612	Internal (High)	DNAT	BGP-Multiline	120.133.15.12	8443	120.133.15.12	8443	tcp	✓	✗	sslvpn
196609	User-defined	SNAT	BGP-Multiline	172.16.0.0/24	Any	120.133.15.12	Any	Any	✓	✗	数据中心外部子网访问外部网络时...
196613	User-defined	DNAT	BGP-Multiline	120.133.15.12	20022	172.16.0.5	22	tcp	✓	✗	
196615	User-defined	DNAT	BGP-Multiline	120.133.15.12	15901	172.16.0.7	5901	tcp	✓	✗	kyligence VNC连接
196622	User-defined	DNAT	BGP-Multiline	120.133.15.12	4200	172.16.0.5	4200	tcp	✓	✗	
196623	User-defined	DNAT	BGP-Multiline	120.133.15.12	8033	172.16.0.5	8033	tcp	✓	✗	

Migration of the production environment from Aliyun ECS to Aliyun ECI

Here are some tips about the migration of the production environment

- ECI allows user to create container basing on given Docker images, the Docker images could come from Aliyun public Docker images, or private repository at Aliyun. We created five private repository at Aliyun: cloudtides (for backend image of CloudTides), cloudtides-frontend (for front end image of CloudTides), scienterprise_server, scienterprise_mysql and scienterprise_makeproject.
- For CloudTides, there are three ECI container group required: backend, frontend and database. For each container group, an Elastic IP Address (EIP) is required. We created three EIP for CloudTides: 106.14.136.120 (for CloudTides backend), 139.224.229.214 (for CloudTides frontend), 47.100.197.116 (for postgreSQL DB). And for the postgreSQL DB container group, we mounted a NAS filesystem 317474bf2f for it to realize data persistence, the directory path is /data/postgres and it is mounted to the path /var/lib/postgresql/data in the container. (Here is a tip for the NAS filesystem, if you want to mount a directory in the NAS filesystem to the ECI container, first you should make sure the directory exist in the NAS filesystem, and the way I used to create a directory is to first mount the NAS filesystem to a ECS server, then create the directory)
- The new version of CI/CD is completed by github actions (https://github.com/cloudtides/CloudTides/blob/master/.github/workflows/backend.yml and https://github.com/cloudtides/CloudTides/blob/master/.github/workflows/frontend.yml). Its workflow is to create the new docker images and push to the corresponding private repository at Aliyun, then restart the corresponding ECI container group with a Aliyun client (code is at https://github.com/cloudtides/CloudTides/tree/master/tides-server/aliyun-script). Remember here you should set the image pulling policy of each container group as "Always", otherwise the new pushed images won't be pulled and restarting the container group will be useless.
- If you accidentally delete the container groups or you have to delete the container group and restart the whole services, you can refer to Launch Templates, there are three templates: cloudtides-backend ,cloudtides-frontend and cloudtides-postgres. You can easily create the backend and frontend ECI container groups with these two templates. But remember when using the templates, you should manually go to the Other Settings part and set up the EIP for each container group see the following figure), and for backend you should choose 106.14.136.120, and for frontend it is 139.224.229.214, and 47.100.197.116 for postgres DB.



Go to the Alibaba Cloud Management Console to apply for a new EIP.

How to realize the template uploading feature for CloudTides (Future Plan)

Here are some plans and tips about how to realize the template uploading feature for CloudTides:

- First we should provide a file server or an object storage to users so they can upload their template file (for vcd it should b OVA file), there are two possible ways for the users to upload files: 1. directly select a local file to upload; 2. provide a URL that can be used to download the file. Here I recommend to choose the first way, because the second way is quite unsafe, users would take the risk to expose their files to the public.
- After template file is uploaded to our file server or object storage, we should be able to check the security of the template, and add our own component (like BOINC client) to make a new template. This part can be manually done or be partially automatic.
- Once the new template is made users should be able to see that their templates are ready to be published from CloudTides dashboard. Then users can select the datacenters they want to publish their templates. For the publishing part, you could refer to the function ***UploadOvf*** at https://github.com/vmware/go-vcloud-director/blob/master/govcd/catalog.go, this function can upload an OVA or OVF file to the specific catalog in a datacenter.

For the file server, one possible solution is Aliyun Object Storage Service (OSS), because CloudTides server is deployed on Aliyun, if the server want to visit the files stored in OSS, it is internal access, and it won't cause any cost. But one significant drawback of this plan is that, after