

멀티 클라우드 서비스 공통 프레임워크 클라우드 드라이버 인터페이스 (v0.1)

Note

- 개발 1단계에서 제공되는 클라우드 드라이버 인터페이스 규격 정의
- 클라우드 자원의 인터페이스 패턴: Create/List/Get/Delete
- 인터페이스 이름의 첫 글자는 대문자 (Go 규칙)

멀티 클라우드 서비스 공통 프레임워크에서 클라우드 드라이버 개발자에게 제공하는 클라우드 드라이버 인터페이스 규격은 다음과 같다.

- ※ 세부 인터페이스 참조: <https://github.com/cloud-barista/poc-cb-spider/tree/master/cloud-driver/interfaces>
- ※ 드라이버 개발 예시 참조: <https://github.com/cloud-barista/poc-cb-spider/tree/master/cloud-driver/drivers/test-a-driver>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
<p>클라우드 드라이버 인터페이스: 클라우드 드라이버의 버전, 제공 기능 및 대상 클라우드 인프라 연결 인터페이스 CloudConnection을 제공</p> <p>※ ConnectCloud() 구현 시 클라우드 연결 정보를 CloudConnection 구현 객체로 전달</p>	<pre>import (icon "github.com/cloud-barista/poc-cb-spider/cloud-driver/interfaces/connect") type DriverCapabilityInfo struct { ImageHandler bool // support: true, do not support: false VNetworkHandler bool // support: true, do not support: false SecurityHandler bool // support: true, do not support: false KeyPairHandler bool // support: true, do not support: false VNicHandler bool // support: true, do not support: false PublicIPHandler bool // support: true, do not support: false VMHandler bool // support: true, do not support: false }</pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> type CredentialInfo struct { // @todo TBD // key-value pairs } type RegionInfo struct { Region string Zone string } type ConnectionInfo struct { CredentialInfo CredentialInfo RegionInfo RegionInfo } type CloudDriver interface { GetDriverVersion() string GetDriverCapability() DriverCapabilityInfo ConnectCloud(connectionInfo ConnectionInfo) (icon.CloudConnection, error) } </pre>
<p>클라우드 연결 인터페이스: 클라우드 인프라 자원별 핸들러 인터페이스 제공</p> <p>※ CreateImageHandler() 등 구현 시 클라우드 연결 정보를 ImageHandler 구현 객체로 전달</p>	<pre> import (irs "github.com/cloud-barista/poc-cb-spider/cloud-driver/interfaces/resources") type CloudConnection interface { CreateImageHandler() (irs.ImageHandler, error) } </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> CreateVNetworkHandler() (irs.VNetworkHandler, error) CreateSecurityHandler() (irs.SecurityHandler, error) CreateKeyPairHandler() (irs.KeyPairHandler, error) CreateVNicHandler() (irs.VNicHandler, error) CreatePublicIPHandler() (irs.PublicIPHandler, error) CreateVMHandler() (irs.VMHandler, error) IsConnected() (bool, error) Close() error } </pre>
<p>클라우드 인프라 자원 핸들러 인터페이스: 클라우드 인프라 자원별로 제어 가능한 핸들러 인터페이스 제공 (예시: ImageHandler 등)</p>	<pre> type ImageReqInfo struct { name string // @todo } type ImageInfo struct { name string id string // @todo } type ImageHandler interface { CreateImage(imageReqInfo ImageReqInfo) (ImageInfo, error) } </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> ListImage() ([]*ImageInfo, error) GetImage(imageID string) (ImageInfo, error) DeleteImage(imageID string) (bool, error) } </pre>
	<pre> type VNetworkReqInfo struct { name string // @todo } type VNetworkInfo struct { name string id string // @todo } type VNetworkHandler interface { CreateVNetwork(vNetworkReqInfo VNetworkReqInfo) (VNetworkInfo, error) ListVNetwork() ([]*VNetworkInfo, error) GetVNetwork(vNetworkID string) (VNetworkInfo, error) DeleteVNetwork(vNetworkID string) (bool, error) } </pre>
	<pre> type SecurityReqInfo struct { name string // @todo } </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> } type SecurityInfo struct { name string id string // @todo } type SecurityHandler interface { CreateSecurity(securityReqInfo SecurityReqInfo) (SecurityInfo, error) ListSecurity() ([]*SecurityInfo, error) GetSecurity(securityID string) (SecurityInfo, error) DeleteSecurity(securityID string) (bool, error) } type KeyPairReqInfo struct { name string // @todo } type KeyPairInfo struct { name string id string // @todo } type KeyPairHandler interface { </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> CreateKey(keyPairReqInfo KeyPairReqInfo) (KeyPairInfo, error) ListKey() ([]*KeyPairInfo, error) GetKey(keyPairID string) (KeyPairInfo, error) DeleteKey(keyPairID string) (bool, error) } </pre>
	<pre> type VNicReqInfo struct { name string // @todo } type VNicInfo struct { name string id string // @todo } type VNicHandler interface { CreateVNic(vNicReqInfo VNicReqInfo) (VNicInfo, error) ListVNic() ([]*VNicInfo, error) GetVNic(vNicID string) (VNicInfo, error) DeleteVNic(vNicID string) (bool, error) } </pre>
	<pre> type PublicIPReqInfo struct { name string } </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> // @todo } type PublicIPInfo struct { name string id string // @todo } type PublicIPHandler interface { CreatePublicIP(publicIPReqInfo PublicIPReqInfo) (PublicIPInfo, error) ListVNetwork() ([]*PublicIPInfo, error) GetVNetwork(publicIPID string) (PublicIPInfo, error) DeleteVNetwork(publicIPID string) (bool, error) } type VMReqInfo struct { // region/zone: Do not specify, this driver already knew these in Connection. name string imageInfo ImageInfo vNetworkInfo VNetworkInfo securityInfo SecurityInfo keyPairInfo KeyPairInfo specID string // instance type or flavour, etc... vNicInfo VNicInfo </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> publicIPInfo PublicIPInfo } // GO do not support Enum. So, define like this. type VMStatus string const (pending VMStatus = "PENDING" // from launch, suspended to running running VMStatus = "RUNNING" suspending VMStatus = "SUSPENDING" // from running to suspended suspended VMStatus = "SUSPENDED" rebooting VMStatus = "REBOOTING" // from running to running termiating VMStatus = "TERMINATING" // from running, suspended to terminated terminated VMStatus = "TERMINATED") type RegionInfo struct { region string zone string } type VMInfo struct { name string </pre>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<div>id string</div> <div>startTime time.Time // Timezone: based on cloud-barista server location.</div> <div>region RegionInfo // ex) {us-east1, us-east1-c} or {ap-northeast-2}</div> <div>imageID string // ex) ami-047f7b46bd6dd5d84 or projects/gce-uefi-images/global/images/centos-7-v20190326</div> <div>specID string // instance type or flavour, etc... ex) t2.micro or f1-micro</div> <div>vNetworkID string // ex) vpc-23ed0a4b</div> <div>subNetworkID string // ex) subnet-8c4a53e4</div> <div>securityID string // ex) sg-0b7452563e1121bb6</div> <div>vNIC string // ex) eth0</div> <div>publicIP string // ex) 13.125.43.21</div> <div>publicDNS string // ex) ec2-13-125-43-0.ap-northeast-2.compute.amazonaws.com</div> <div>privateIP string // ex) ip-172-31-4-60.ap-northeast-2.compute.internal</div> <div>privateDNS string // ex) 172.31.4.60</div> <div>keyPairID string // ex) powerkimKeyPair</div> <div>guestUserID string // ex) user1</div> <div>guestUserPwd string</div> <div>guestBootDisk string // ex) /dev/sda1</div>

분류	클라우드 드라이버 인터페이스 및 관련 데이터 자료형(Go Language Spec)
	<pre> guestBlockDisk string // ex) additionalInfo string // Any information to be good for users and developers. } type VMHandler interface { StartVM(vmReqInfo VMReqInfo) (VMInfo, error) SuspendVM(vmID string) ResumeVM(vmID string) RebootVM(vmID string) TerminateVM(vmID string) ListVMStatus() []*VMStatus GetVMStatus(vmID string) VMStatus ListVM() []*VMInfo GetVM(vmID string) VMInfo } </pre>