Enable azure ingress with lets encrypt -Aka application routing.

* Run helm init to install tiller.
* Install cert manager :

helm install stable/cert-manager \

--set ingressShim.defaultIssuerName=letsencrypt-staging \

--set ingressShim.defaultIssuerKind=ClusterIssuer \

--set rbac.create=false \

--set serviceAccount.create=false

* Install cluster issuer for staging.

apiVersion: certmanager.k8s.io/v1alpha1

kind: ClusterIssuer

metadata:

name: letsencrypt-staging

spec:

acme:

server: https://acme-staging-v02.api.letsencrypt.org/directory

email: dinorg@gmail.com

privateKeySecretRef:

name: letsencrypt-staging

http01: {}

* Wait for the issuer to be registered :

Name: letsencrypt-staging

Namespace:

Labels: <none>

Annotations: <none>

API Version: certmanager.k8s.io/v1alpha1

Kind: ClusterIssuer

Metadata:

Cluster Name:

Creation Timestamp: 2018-07-26T16:53:53Z

Generation: 1

Resource Version: 6615

Self Link: /apis/certmanager.k8s.io/v1alpha1/clusterissuers/letsencrypt-staging

UID: 7a39243c-90f4-11e8-a04a-e2d2f18309ea

Spec:

Acme:

Email: dinorg@gmail.com

Http 01:

Private Key Secret Ref:

Key:

Name: letsencrypt-staging

Server: https://acme-staging-v02.api.letsencrypt.org/directory

Status:

Acme:

Uri: https://acme-staging-v02.api.letsencrypt.org/acme/acct/6536052

Conditions:

Last Transition Time: 2018-07-26T16:53:54Z

Message: The ACME account was registered with the ACME server

Reason: ACMEAccountRegistered

Status: True

Type: Ready

Events: <none>

* Create certificate object : make sure you change the ingress class to the below:

apiVersion: certmanager.k8s.io/v1alpha1

kind: Certificate

metadata:

name: tls-secret

spec:

secretName: tls-secret

dnsNames:

- dinor-aks-ingress.secured.ce26b10d62d84ffd9866.westus.aksapp.io

acme:

config:

- http01:

**ingressClass: addon-http-application-routing**

domains:

- dinor-aks-ingress.secured.ce26b10d62d84ffd9866.westus.aksapp.io

issuerRef:

name: letsencrypt-staging

kind: ClusterIssuer

* Create ingress :

Make sure to change the fields in yellow :

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

name: hello-world-prod-non-prod

annotations:

kubernetes.io/ingress.class: addon-http-application-routing

certmanager.k8s.io/cluster-issuer: letsencrypt-staging

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

tls:

- hosts:

- dinort-test.1bbd39221c62448a9228.westus.aksapp.io

secretName: tls-secret

rules:

- host: dinort-test.1bbd39221c62448a9228.westus.aksapp.io

http:

paths:

- path: /

backend:

serviceName: aks-helloworld

servicePort: 80

* Add application :

Check certmanager logs :

E0726 16:53:55.395361 1 controller.go:147] clusterissuers controller: Re-queuing item "letsencrypt-staging" due to error processing: Operation cannot be fulfilled on clusterissuers.certmanager.k8s.io "letsencrypt-staging": the object has been modified; please apply your changes to the latest version and try again

I0726 16:53:55.395402 1 controller.go:138] clusterissuers controller: syncing item 'letsencrypt-staging'

I0726 16:53:55.395443 1 acme.go:162] getting private key (letsencrypt-staging->tls.key) for acme issuer default/letsencrypt-staging

I0726 16:53:55.395809 1 logger.go:67] Calling GetAccount

I0726 16:53:55.953999 1 setup.go:73] letsencrypt-staging: verified existing registration with ACME server

I0726 16:53:55.954049 1 controller.go:152] clusterissuers controller: Finished processing work item "letsencrypt-staging"

I0726 16:53:55.954063 1 controller.go:138] clusterissuers controller: syncing item 'letsencrypt-staging'

I0726 16:53:55.954091 1 acme.go:162] getting private key (letsencrypt-staging->tls.key) for acme issuer default/letsencrypt-staging

I0726 16:53:55.954427 1 logger.go:67] Calling GetAccount

I0726 16:53:56.462872 1 setup.go:73] letsencrypt-staging: verified existing registration with ACME server

I0726 16:53:56.462944 1 controller.go:152] clusterissuers controller: Finished processing work item "letsencrypt-staging"

If you don’t see any progress restart ingress pod :and check the logs again :

* If still no progress ,kill the certmanager pod :
* Look at the logs :

1 prepare.go:263] Cleaning up old/expired challenges for Certificate default/tls-secret

I0726 17:16:40.706631 1 logger.go:47] Calling GetChallenge

I0726 17:16:40.836141 1 ingress.go:33] Looking up Ingresses for selector certmanager.k8s.io/acme-http-domain=2466320436,certmanager.k8s.io/acme-http-token=682429914

I0726 17:16:40.836181 1 helpers.go:188] Found status change for Certificate "tls-secret" condition "Ready": "False" -> "False"; setting lastTransitionTime to 2018-07-26 17:16:40.836176321 +0000 UTC m=+40.442701415

I0726 17:16:40.836195 1 sync.go:261] Error preparing issuer for certificate default/tls-secret: http-01 self check failed for domain "dinort-test.1bbd39221c62448a9228.westus.aksapp.io"

E0726 17:16:40.836210 1 sync.go:182] [default/tls-secret] Error getting certificate 'tls-secret': secret "tls-secret" not found

I0726 17:16:40.892351 1 controller.go:152] ingress-shim controller: syncing item 'default/hello-world-prod-non-prod'

I0726 17:16:40.892382 1 sync.go:124] Certificate "tls-secret" for ingress "hello-world-prod-non-prod" already exists

I0726 17:16:40.892410 1 sync.go:127] Certificate "tls-secret" for ingress "hello-world-prod-non-prod" is up to date

I0726 17:16:40.892419 1 controller.go:166] ingress-shim controller: Finished processing work item "default/hello-world-prod-non-prod"

E0726 17:16:40.894467 1 controller.go:186] certificates controller: Re-queuing item "default/tls-secret" due to error processing: http-01 self check failed for domain "dinort-test.1bbd39221c62448a9228.westus.aksapp.io"

The above is normal.

Wait for the new ing rules to appear :

cm-acme-http-solver-vqsr5 dinort-test.1bbd39221c62448a9228.westus.aksapp.io 104.42.158.62 80 2m

cm-acme-http-solver-wttvb dinor-test.1bbd39221c62448a9228.westus.aksapp.io 104.42.158.62 80 2m

this means that acme is now creating a pod for validation :

cm-acme-http-solver-554fd 1/1 Running 0 3m

cm-acme-http-solver-bcltk 1/1 Running 0 3m

* Wait for 10 min ,if still the same issue :

Do describe on one of the ing rule and try to reach the website.

If you reach it and get the token then all is ok.

* Make sure the cert is issued :

kubectl get certificate -o yaml

apiVersion: v1

items:

- apiVersion: certmanager.k8s.io/v1alpha1

kind: Certificate

metadata:

clusterName: ""

creationTimestamp: 2018-07-26T17:16:16Z

generation: 1

name: tls-secret

namespace: default

ownerReferences:

- apiVersion: extensions/v1beta1

blockOwnerDeletion: true

controller: true

kind: Ingress

name: hello-world-prod-non-prod

uid: 6aa2d334-90f6-11e8-a04a-e2d2f18309ea

resourceVersion: "9974"

selfLink: /apis/certmanager.k8s.io/v1alpha1/namespaces/default/certificates/tls-secret

uid: 9aadba35-90f7-11e8-a04a-e2d2f18309ea

spec:

acme:

config:

- domains:

- dinort-test.1bbd39221c62448a9228.westus.aksapp.io

http01:

ingress: ""

ingressClass: addon-http-application-routing

commonName: ""

dnsNames:

- dinort-test.1bbd39221c62448a9228.westus.aksapp.io

issuerRef:

kind: ClusterIssuer

name: letsencrypt-staging

secretName: tls-secret

status:

acme:

order:

url: https://acme-staging-v02.api.letsencrypt.org/acme/order/6536052/4846644

conditions:

- lastTransitionTime: 2018-07-26T17:23:02Z

message: Certificate issued successfully

reason: CertIssued

status: "True"

type: Ready

- lastTransitionTime: 2018-07-26T17:23:01Z

message: Order validated

reason: OrderValidated

status: "False"

type: ValidateFailed

- apiVersion: certmanager.k8s.io/v1alpha1

kind: Certificate

metadata:

clusterName: ""

creationTimestamp: 2018-07-26T17:00:28Z

generation: 1

name: tls-secret-non-prod

namespace: default

resourceVersion: "9839"

selfLink: /apis/certmanager.k8s.io/v1alpha1/namespaces/default/certificates/tls-secret-non-prod

uid: 65b19036-90f5-11e8-a04a-e2d2f18309ea

spec:

acme:

config:

- domains:

- dinor-test.1bbd39221c62448a9228.westus.aksapp.io

http01:

ingress: ""

ingressClass: addon-http-application-routing

commonName: ""

dnsNames:

- dinor-test.1bbd39221c62448a9228.westus.aksapp.io

issuerRef:

kind: ClusterIssuer

name: letsencrypt-staging

secretName: tls-secret

status:

acme:

order:

url: https://acme-staging-v02.api.letsencrypt.org/acme/order/6536052/4846333

conditions:

- lastTransitionTime: 2018-07-26T17:21:58Z

message: Certificate issued successfully

reason: CertIssued

status: "True"

type: Ready

- lastTransitionTime: null

message: Order validated

reason: OrderValidated

status: "False"

type: ValidateFailed

kind: List

metadata:

resourceVersion: ""

selfLink: ""

* Make sure you got the keys also :

dinorgeler@Dinors-MBP: local$kubectl get secret

NAME TYPE DATA AGE

default-token-jvgbq kubernetes.io/service-account-token 3 1h

letsencrypt-staging Opaque 1 32m

tls-secret kubernetes.io/tls 2 4m

* Check if the url is reachable :

You could end up with default backend :

curl https://dinort-test.1bbd39221c62448a9228.westus.aksapp.io --insecure

default backend – 404

in that case :

* Goto to the azure ingress and get the logs :

W0726 17:21:55.700653 6 controller.go:775] service default/cm-acme-http-solver-st22d does not have any active endpoints

W0726 17:21:55.700730 6 controller.go:1035] ssl certificate "default/tls-secret" does not exist in local store

I0726 17:21:55.700758 6 controller.go:170] backend reload required

I0726 17:21:55.814870 6 event.go:218] Event(v1.ObjectReference{Kind:"Ingress", Namespace:"default", Name:"cm-acme-http-solver-vqsr5", UID:"9cbeeed1-90f7-11e8-a04a-e2d2f18309ea", APIVersion:"extensions", ResourceVersion:"9815", FieldPath:""}): type: 'Normal' reason: 'DELETE' Ingress default/cm-acme-http-solver-vqsr5

I0726 17:21:55.970433 6 controller.go:179] ingress backend successfully reloaded...

I0726 17:21:56.031138 6 event.go:218] Event(v1.ObjectReference{Kind:"Ingress", Namespace:"default", Name:"cm-acme-http-solver-wttvb", UID:"9c803486-90f7-11e8-a04a-e2d2f18309ea", APIVersion:"extensions", ResourceVersion:"9826", FieldPath:""}): type: 'Normal' reason: 'DELETE' Ingress default/cm-acme-http-solver-wttvb

W0726 17:21:59.034206 6 controller.go:1035] ssl certificate "default/tls-secret" does not exist in local store

I0726 17:21:59.034256 6 controller.go:170] backend reload required

I0726 17:21:59.223685 6 controller.go:179] ingress backend successfully reloaded...

W0726 17:22:02.367457 6 controller.go:1035] ssl certificate "default/tls-secret" does not exist in local store

W0726 17:24:42.388694 6 backend\_ssl.go:48] error obtaining PEM from secret default/tls-secret: error retrieving secret default/tls-secret: secret default/tls-secret was not found

W0726 17:24:42.388837 6 controller.go:1035] ssl certificate "default/tls-secret" does not exist in local store

Look for your ingress rule ,if its not there just kill the ingress pod and look at the logs once again :

I0726 17:38:31.062768 7 status.go:356] updating Ingress default/hello-world-prod-non-prod status to [{104.42.158.62 }]

I0726 17:38:31.085704 7 event.go:218] Event(v1.ObjectReference{Kind:"Ingress", Namespace:"default", Name:"hello-world-prod-non-prod", UID:"6aa2d334-90f6-11e8-a04a-e2d2f18309ea", APIVersion:"extensions", ResourceVersion:"11697", FieldPath:""}): type: 'Normal' reason: 'UPDATE' Ingress default/hello-world-prod-non-prod

Then you have to wait for the dns to propgate.

* Check the dns :

Default Server: UnKnown

Address: 40.90.4.7

> dinort-test.1bbd39221c62448a9228.westus.aksapp.io

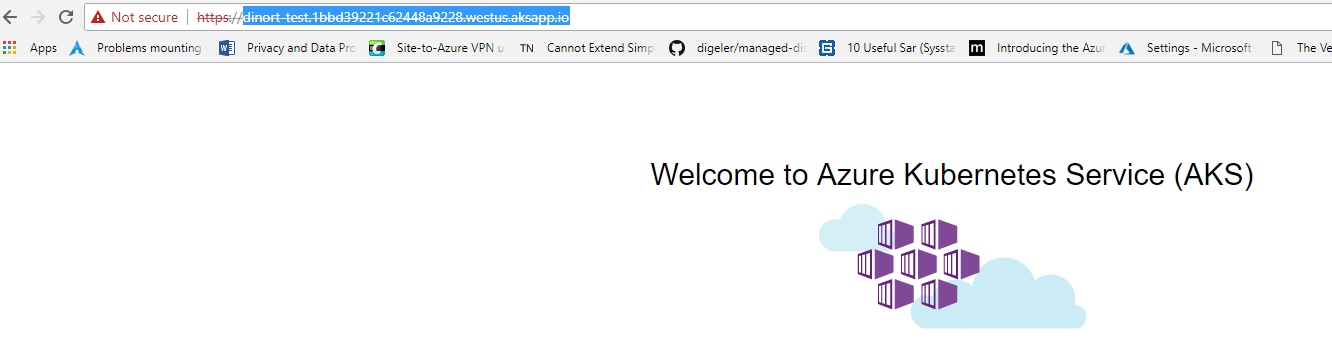
Server: UnKnown

Address: 40.90.4.7

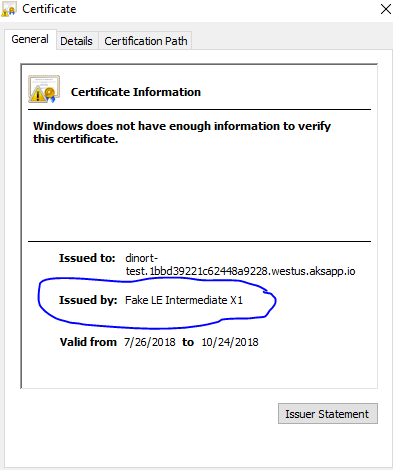
Name: dinort-test.1bbd39221c62448a9228.westus.aksapp.io

Address: 104.42.158.62

* Now try to browse :



Check the fake cert :



Moving to prod cert :

* First create new issuer :

apiVersion: certmanager.k8s.io/v1alpha1

kind: ClusterIssuer

metadata:

name: letsencrypt-prod

spec:

acme:

server: https://acme-v02.api.letsencrypt.org/directory

email: dinorg@gmail.com

privateKeySecretRef:

name: letsencrypt-prod

http01: {}

apply the file

* Check that you are registered :

local$kubectl get clusterissuer letsencrypt-prod-for-dinor -o yaml

apiVersion: certmanager.k8s.io/v1alpha1

kind: ClusterIssuer

metadata:

clusterName: ""

creationTimestamp: 2018-07-26T17:55:56Z

generation: 1

name: letsencrypt-prod-for-dinor

namespace: ""

resourceVersion: "13643"

selfLink: /apis/certmanager.k8s.io/v1alpha1/clusterissuers/letsencrypt-prod-for-dinor

uid: 256dfb80-90fd-11e8-a04a-e2d2f18309ea

spec:

acme:

email: dinorg@gmail.com

http01: {}

privateKeySecretRef:

key: ""

name: letsencrypt-prod-for-dinor

server: https://acme-v02.api.letsencrypt.org/directory

status:

acme:

uri: https://acme-v02.api.letsencrypt.org/acme/acct/39059544

conditions:

- lastTransitionTime: 2018-07-26T17:55:58Z

message: The ACME account was registered with the ACME server

reason: ACMEAccountRegistered

status: "True"

type: Ready

* Create the certificate object :

apiVersion: certmanager.k8s.io/v1alpha1

kind: Certificate

metadata:

name: tls-secret-prod

spec:

secretName: tls-secret-prod

dnsNames:

- dinort-prod.1bbd39221c62448a9228.westus.aksapp.io

acme:

config:

- http01:

ingressClass: addon-http-application-routing

domains:

- dinort-prod.1bbd39221c62448a9228.westus.aksapp.io

issuerRef:

name: letsencrypt-prod-for-dinor

kind: ClusterIssuer

* Create ingress :

apiVersion: extensions/v1beta1

kind: Ingress

metadata:

name: hello-world-prod

annotations:

kubernetes.io/ingress.class: addon-http-application-routing

certmanager.k8s.io/cluster-issuer: letsencrypt-prod-for-dinor

nginx.ingress.kubernetes.io/rewrite-target: /

spec:

tls:

- hosts:

- dinord-prod.1bbd39221c62448a9228.westus.aksapp.io

secretName: tls-secret-prod-dinor

rules:

- host: dinor-prod.1bbd39221c62448a9228.westus.aksapp.io

http:

paths:

- path: /

backend:

serviceName: aks-helloworld

servicePort: 80

* Check ingress logs :

0726 18:11:03.898770 7 backend\_ssl.go:48] error obtaining PEM from secret default/tls-secret-prod-dinor: error retrieving secret default/tls-secret-prod-dinor: secret default/tls-secret-prod-dinor was not found

W0726 18:11:03.898949 7 controller.go:1035] ssl certificate "default/tls-secret-prod-dinor" does not exist in local store

I0726 18:11:03.898988 7 controller.go:170] backend reload required

I0726 18:11:04.387658 7 controller.go:179] ingress backend successfully reloaded...

I0726 18:11:31.055698 7 status.go:356] updating Ingress default/hello-world-prod status to [{104.42.158.62 }]

I0726 18:11:31.153782 7 event.go:218] Event(v1.ObjectReference{Kind:"Ingress", Namespace:"default", Name:"hello-world-prod", UID:"42565334-90ff-11e8-a04a-e2d2f18309ea", APIVersion:"extensions", ResourceVersion:"15349", FieldPath:""}): type: 'Normal' reason: 'UPDATE' Ingress default/hello-world-prod

W0726 18:11:31.153805 7 backend\_ssl.go:48] error obtaining PEM from secret default/tls-secret-prod-dinor: error retrieving secret default/tls-secret-prod-dinor: secret default/tls-secret-prod-dinor was not found

W0726 18:11:31.153919 7 controller.go:1035] ssl certificate "default/tls-secret-prod-dinor" does not exist in local store

* Check certmanager logs ,if you see old records you will have to kill the certmanager pod :

I0726 17:55:58.855850 1 controller.go:152] clusterissuers controller: Finished processing work item "letsencrypt-prod-for-dinor"

I0726 17:55:58.855864 1 controller.go:138] clusterissuers controller: syncing item 'letsencrypt-prod-for-dinor'

I0726 17:55:58.855892 1 acme.go:162] getting private key (letsencrypt-prod-for-dinor->tls.key) for acme issuer default/letsencrypt-prod-for-dinor

I0726 17:55:58.856223 1 logger.go:67] Calling GetAccount

I0726 17:55:59.297567 1 setup.go:73] letsencrypt-prod-for-dinor: verified existing registration with ACME server

I0726 17:55:59.297613 1 controller.go:152] clusterissuers controller: Finished processing work item "letsencrypt-prod-for-dinor"

* then check the logs again

0726 18:24:35.004621 1 logger.go:72] Calling HTTP01ChallengeResponse

I0726 18:24:35.004669 1 prepare.go:263] Cleaning up old/expired challenges for Certificate default/tls-secret-prod-dinor

I0726 18:24:35.004683 1 logger.go:47] Calling GetChallenge

I0726 18:24:35.134243 1 ingress.go:33] Looking up Ingresses for selector certmanager.k8s.io/acme-http-domain=2068910005,certmanager.k8s.io/acme-http-token=1207307941

I0726 18:24:35.134302 1 helpers.go:188] Found status change for Certificate "tls-secret-prod-dinor" condition "Ready": "False" -> "False"; setting lastTransitionTime to 2018-07-26 18:24:35.134298003 +0000 UTC m=+462.565764364

I0726 18:24:35.134327 1 sync.go:261] Error preparing issuer for certificate default/tls-secret-prod-dinor: http-01 self check failed for domain "dinor-prod.1bbd39221c62448a9228.westus.aksapp.io"

E0726 18:24:35.134340 1 sync.go:182] [default/tls-secret-prod-dinor] Error getting certificate 'tls-secret-prod-dinor': secret "tls-secret-prod-dinor" not found

E0726 18:24:35.167380 1 controller.go:186] certificates controller: Re-queuing item "default/tls-secret-prod-dinor" due to error processing: http-01 self check failed for domain "dinor-prod.1bbd39221c62448a9228.westus.aksapp.io"

The above is normal we need to check if ingress rule is created by acme .

* $kubectl get ing
* NAME HOSTS ADDRESS PORTS AGE
* cm-acme-http-solver-65zwk dinor-prod.1bbd39221c62448a9228.westus.aksapp.io 104.42.158.62 80 5m
* cm-acme-http-solver-9r2cd dinort-prod.1bbd39221c62448a9228.westus.aksapp.io 104.42.158.62 80 8m
* hello-world-prod dinor-prod.1bbd39221c62448a9228.westus.aksapp.io 104.42.158.62 80, 443 14m
* hello-world-prod-non-prod dinort-test.1bbd39221c62448a9228.westus.aksapp.io 104.42.158.62 80, 443 1h
* check if we can reach the challenge if yes give it 5 min then check the cert

kubectl get certificate tls-secret-prod-dinor -o yaml

apiVersion: certmanager.k8s.io/v1alpha1

kind: Certificate

metadata:

clusterName: ""

creationTimestamp: 2018-07-26T18:08:07Z

generation: 1

name: tls-secret-prod-dinor

namespace: default

resourceVersion: "17434"

selfLink: /apis/certmanager.k8s.io/v1alpha1/namespaces/default/certificates/tls-secret-prod-dinor

uid: d9658dd0-90fe-11e8-a04a-e2d2f18309ea

spec:

acme:

config:

- domains:

- dinor-prod.1bbd39221c62448a9228.westus.aksapp.io

http01:

ingress: ""

ingressClass: addon-http-application-routing

commonName: ""

dnsNames:

- dinor-prod.1bbd39221c62448a9228.westus.aksapp.io

issuerRef:

kind: ClusterIssuer

name: letsencrypt-prod-for-dinor

secretName: tls-secret-prod-dinor

status:

acme:

order:

url: ""

conditions:

- lastTransitionTime: 2018-07-26T18:27:48Z

message: 'Failed to finalize order: acme: urn:ietf:params:acme:error:rateLimited:

Error finalizing order :: too many certificates already issued for exact set

of domains: dinor-prod.1bbd39221c62448a9228.westus.aksapp.io: see https://letsencrypt.org/docs/rate-limits/'

reason: IssueError

status: "False"

type: Ready

- lastTransitionTime: 2018-07-26T18:28:33Z

message: Order validated

reason: OrderValidated

status: "False"

type: ValidateFailed

make sure you got the secret :

default-token-jvgbq kubernetes.io/service-account-token 3 2h

letsencrypt-prod Opaque 1 38m

letsencrypt-prod-for-dinor Opaque 1 34m

letsencrypt-staging Opaque 1 1h

tls-secret kubernetes.io/tls 2 1h

tls-secret-prod-dinor kubernetes.io/tls 2 4m

check that you can browse the website :

