kops requires --ssh-public-key even if I want to use an existing one #4728

Closed **mludvig** opened this issue on Mar 20, 2018 · 37 comments

Labels lifecycle/rotten

mludvig commented on Mar 20, 2018

For AWS deployments *kops v1.9.0-alpha1* requires `--ssh-public-key` even if we already have an existing key pair that we would like to use. All our EC2 instances must use the same SSH key and I am actually not permitted to create a new key pair at all in the prod account (I could change my IAM permissions but that's beyond the point).

I propose to add a new option `--ssh-key-name={existing-keypair-name}` and if that is supplied don't require `--ssh-public-key` and don't attempt to create a new key pair.

I believe this should be a relatively simple change but I'm not a *go* programmer so can't do it myself, sorry.

Hope someone can help with that! :)

8 1

ihac commented on Mar 24, 2018 Contributor

Hi, **@mludvig**, I've been developing kops for a while, and as far as I know, kops already allows us to use existing key pair with option `---ssh-public-key <public_key_file>` . By default, kops uses `~/.ssh/id_rsa.pub` as the default key and publishes it to your instances automatically.

I'm not sure if I've made it clear, or maybe you could provide more details about in which situations kops won't work as expected.

1

mludvig commented on Mar 24, 2018 • edited Author

Hi **@ihac**, with `--ssh-public-key` kops creates a new EC2 keypair in AWS. Even if it takes your existing locally created key and uploads to AWS it's still a "new" key in AWS.

Our organisation rules don't allow that, we've got a pre-created "official" EC2 keypair in each AWS account, named e.g. *aws-sandpit*, and we are supposed to use that for all EC2 instances that we create.

Hence I need a switch like `--ssh-key-name aws-sandpit` what will make kops skip the EC2 key creation and instead create the instances with the existing *aws-sandpit* keypair.

I noticed that the cluster_spec.yaml already supports that (https://github.com/kubernetes/kops/blob/master/docs/cluster_spec.md#sshkeyname) - all I need now is to expose the same functionality in a command line switch like `--ssh-key-name` .

Do you think it's possible?

👍 6 ❤️ 2

mludvig commented on Mar 24, 2018 **Author**

Hi again **@ihac**
I tried to do it myself a few days ago but as I don't know neither Go nor Kops internal I didn't get very far. Here's my attempt to date, hope you can take it from there :)
ssh-key-name-attempt.diff.txt

ihac commented on Mar 24, 2018 **Contributor**

Seems that you wanna use the ssh key pre-created on AWS, rather than those on your local computer, is that right?

For what I know, kops currently does not support users to set the `SSHKeyName` field in cluster spec, and it would upload the local ssh key to aws and name it with `kubernetes.<cluster_name>-<publickey_fingerprint>` . This might be the "new" key you mentioned?

Back to your problem, I think there are two solutions:

1. Download the pre-created keypair(e.g. aws-sandpit) to your computer, and set it by `--ssh-public-key` when creating a new cluster. In this way, kops would still create a "new" keypair, but the contents should be the same.
2. Add a new option `--ssh-key-name` for kops. I'll try to work on this with a fork but cannot make a promise, since I'm developing and testing on Alibaba Cloud in China, not AWS. Anyway, I'll ping you for any useful updates.

👍 2 ❤️ 2

chrislovecnm commented on Mar 25, 2018 **Contributor**

We do have https://github.com/kubernetes/kops/blob/master/docs/secrets.md
Which shows how to add a key, also I am fairly certain we can reuse one.

The problem is that the cli at times expects an ssh key locally still. We need to do some tweaks so that if you are re-using a key we do not force one localy

❤️ 1

chrislovecnm commented on Mar 25, 2018 Contributor

Reuse one == reuse an existing key in aws.

❤️ 3

mludvig commented on Mar 26, 2018 Author

Hi **@chrislovecnm** this is what works without creating a new AWS key:

1. First **create cluster** config in S3 but don't specify `--yes`

```
~ $ kops-1.9.0-alpha.2 create cluster --cloud aws [..all the options..] \
        --name k8keyname.example.com
```

2. Next **edit cluster** and add `sshKeyName: aws-sandpit` under `.spec` section, where `aws-sandpit` is an already existing EC2 SSH Key Pair.

```
~ $ kops-1.9.0-alpha.2 edit cluster k8keyname.example.com
```

3. Finally call **update cluster** to actually create it with `--yes`

```
~ $ kops-1.9.0-alpha.2 update cluster k8keyname.example.com --yes
```

This sequence will create the cluster with all instances using `aws-sandpit` keypair instead of creating a new one from `~/.ssh/id_rsa.pub`. Exactly what I wanted! :)

All I'm after now is the ability to combine all these steps together into a single `create cluster` command with `--ssh-key-name` specified, instead of having to go through the unintuitive 3-step process. Something like:

```
kops create cluster [...] --ssh-key-name aws-sandpit --yes
```

My half-baked patch has been attached above but I don't know how to turn the command line parameter into spec file attribute. Hope someone can help with that...

16

jhohertz commented on Apr 11, 2018 Contributor

Just wanted to leave a note as I'm experiencing something similar, where we are generating a YAML through another process to feed to `kops create cluster`. In that case it is insufficient to specify the sshKeyName in the spec section. Upon trying to run the update, it will complain that I have not yet created a secret.

For now I am just going to extract the key via awscli or boto to make it happy, but it would be nice if we could just specify the aws key name in the attribute and let kops handle things from there.

1

xydinesh commented on Jul 7, 2018 • edited

Until this issue is fixed, you can extract public key from AWS keypair and use it with `--ssh-public-key` option.
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-key-pairs.html#retrieving-the-public-key

amitsaha commented on Sep 10, 2018

I have got an implementation of this now which is only tested for AWS, plan to make a PR after cleanup.

1 7

maver1ck commented on Oct 25, 2018

Hi,
I've got a question regarding this topic.
I made kops to use existing keypair by specifing sshKeyName in cluster spec.
But why kops want to create also sshpublickey secret ?

```
SSH public key must be specified when running with AWS (create with `kops create secret --na
cluster_name.com sshpublickey admin -i ~/.ssh/id_rsa.pub`)
```

rjanovski commented on Jan 3, 2019

yes, **@maver1ck** , it seems to be the case when you run create with --dry-run. it does not create the secrets it needs and complains...

try to run kops create without dry-run/saving to file, and then get the file after it was created:
kops get --name=$NAME -oyaml > cluster.yaml
then the secrets were created in s3 and no complaints at kops update :)

maver1ck commented on Jan 8, 2019

But I'm creating cluster using `kops replace -f cluster.yaml --force`
And this is not working

rjanovski commented on Jan 8, 2019

First create, then export yaml, edit and replace. Replace by itself does not seem to initialize all resources, like the necessary secrets.

maver1ck commented on Jan 8, 2019 • edited

@rjanovski
I'm using it to automate K8S cluster creation.
I have cluster.yaml in a git repo and then using replace to create cluster.
I think this is a bug in replace. (it should work similar to create)

rjanovski commented on Jan 8, 2019

@maver1ck did you try this:

```
kops create -f my-cluster.yaml
```

tomaszkiewicz commented on Jan 8, 2019

See my way to manage cluster from a file:

```
kops get cluster --name $KOPS_CLUSTER_NAME

if [ \$? -ne 0 ]; then
  echo "Cluster not found, creating a new one"
  // we don't care about details, only required things to just create a key file in s3
  kops create cluster --name=$KOPS_CLUSTER_NAME --ssh-public-key key.pub --cloud=aws --zones
eu-west-1a
fi

kops replace -f cluster.yaml --force

// then kops update etc
```

slashr commented on Feb 5, 2019 • edited

> But why kops want to create also sshpublickey secret ?

@maver1ck I believe it needs your public key for accessing the bastion server. It adds your public key to the bastion server's authorized_keys directory thus enabling you to SSH to it.

michaelajr commented on Apr 25, 2019

We don't even use ssh keys. Don't have any in the AWS account. AMIs come with AD set up. Is there a way around having to 1) specify a `sshKey` in the YAML, and 2) uploading a dummy one to the state store just to get past the check on `kops update`? This has been an issue for years.

rralcala mentioned this issue on May 7, 2019

Use existing SSHKeyName if no public key is created. #6886

Merged

fejta-bot commented on Jul 24, 2019

Issues go stale after 90d of inactivity.
Mark the issue as fresh with `/remove-lifecycle stale`.
Stale issues rot after an additional 30d of inactivity and eventually close.

If this issue is safe to close now please do so with `/close`.

Send feedback to sig-testing, kubernetes/test-infra and/or fejta.
/lifecycle stale

k8s-ci-robot added the lifecycle/stale label on Jul 24, 2019

fejta-bot commented on Aug 23, 2019

Stale issues rot after 30d of inactivity.
Mark the issue as fresh with `/remove-lifecycle rotten`.
Rotten issues close after an additional 30d of inactivity.

If this issue is safe to close now please do so with `/close`.

Send feedback to sig-testing, kubernetes/test-infra and/or fejta.
/lifecycle rotten

k8s-ci-robot added lifecycle/rotten and removed lifecycle/stale labels on Aug 23, 2019

michaelajr commented on Aug 26, 2019

/remove-lifecycle rotten

k8s-ci-robot removed the lifecycle/rotten label on Aug 26, 2019

michaelajr commented on Aug 26, 2019

Work has been done on this. #7096

fejta-bot commented on Nov 24, 2019

Issues go stale after 90d of inactivity.
Mark the issue as fresh with `/remove-lifecycle stale`.
Stale issues rot after an additional 30d of inactivity and eventually close.

If this issue is safe to close now please do so with `/close`.

Send feedback to sig-testing, kubernetes/test-infra and/or fejta.
/lifecycle stale

k8s-ci-robot added the lifecycle/stale label on Nov 24, 2019

ghost commented on Dec 6, 2019

/remove-lifecycle stale

This would be really beneficial working with aws

k8s-ci-robot removed the lifecycle/stale label on Dec 6, 2019

jhohertz commented on Dec 6, 2019 Contributor

99% sure this bug is no longer relevant, and has been addressed since... a few versions ago (not 100% sure when... I want to say 1.12.x, but I might be a bit off)

ghost commented on Dec 6, 2019

> 99% sure this bug is no longer relevant, and has been addressed since... a few versions ago (not 100% sure when... I want to say 1.12.x, but I might be a bit off)

Is there some way to specify the aws ec2 key to use besides adding a tag to the cluster config? I'm on 1.15 and ran into this yesterday. The only way to get around it was to follow suggestion by **@mludvig** from above.

jhohertz commented on Dec 12, 2019 Contributor

Not sure, My workflow is to always generate and feed the YAML to kops, so re: CLI flags methods, I can't really speak to that.

xunliu commented on Jan 14, 2020

This is a feature we really need!

fejta-bot commented on Apr 13, 2020

Issues go stale after 90d of inactivity.
Mark the issue as fresh with `/remove-lifecycle stale`.
Stale issues rot after an additional 30d of inactivity and eventually close.

If this issue is safe to close now please do so with `/close`.

Send feedback to sig-testing, kubernetes/test-infra and/or fejta.
/lifecycle stale

k8s-ci-robot added the lifecycle/stale label on Apr 13, 2020

fejta-bot commented on May 13, 2020

Stale issues rot after 30d of inactivity.
Mark the issue as fresh with `/remove-lifecycle rotten`.
Rotten issues close after an additional 30d of inactivity.

If this issue is safe to close now please do so with `/close`.

Send feedback to sig-testing, kubernetes/test-infra and/or fejta.
/lifecycle rotten

k8s-ci-robot added lifecycle/rotten and removed lifecycle/stale labels on May 13, 2020

ghost commented on May 13, 2020 via email

/remove-lifecycle rotten

@
...

k8s-ci-robot added lifecycle/rotten and removed lifecycle/rotten labels on May 13, 2020

fejta-bot commented on Jun 12, 2020

Rotten issues close after 30d of inactivity.
Reopen the issue with `/reopen`.
Mark the issue as fresh with `/remove-lifecycle rotten`.

Send feedback to sig-testing, kubernetes/test-infra and/or fejta.
/close

k8s-ci-robot closed this as completed on Jun 12, 2020

---

k8s-ci-robot commented on Jun 12, 2020 Contributor

@fejta-bot: Closing this issue.

▶ Details

olemarkus commented on Jun 12, 2020 Member

Trying to figure out what is left of this issue. Is it being able to create a cluster without ssh keys using `kops create cluster`?
If so, I would perhaps keep it closed as typically one wants to use `kops create -f` when creating a proper production cluster. `kops create cluster` is more of a convenient method to just test kops, but will never give you all the flags you need to fully configure your cluster.

damianobarbati commented on Dec 27, 2022 • edited

@michaelajr as of today the `--ssh-public-key <path>` in kops v1.25.3 is not working with the following combination of parameters:

```
kops create cluster \
  --name $KOPS_CLUSTER_NAME \
  --cloud aws \
```

```
    --cloud-labels 'Project=kops-admin-area' \
    --master-count 1 \
    --master-size m5a.large \
    --master-zones eu-west-1a \
    --node-count 1 \
    --node-size m5a.large \
    --node-volume-size 64 \
    --zones eu-west-1a \
    --networking calico \
    --topology private \
    --ssh-public-key <pub file> \
    --bastion \
    --dry-run \
    -o yaml > $CLUSTER_CONFIG

kops create -f $CLUSTER_CONFIG
kops update cluster --yes
```

Private and pub file was generated using `ssh-keygen -t rsa. In the` config.yml` there is nothing about ssh, and there's no keypair created in aws with the generated keys.
After applying the config, the login to the bastion is denied (using the specified .pub or my local pub, I tried both).

What is the current solution to be able to use either:
A) an existing keypair by name
B) a local keypair created

?

johngmyers commented on Dec 28, 2022 Member

Between the `kops create -f` and the `kops update cluster` you'd need to add `kops create sshpublickey -i <pub file>`

We could probably define an sshkey resource for the dryrun to emit and the create to consume, but that's not this issue.

Assignees

No one assigned

Labels

lifecycle/rotten

Projects

None yet

Milestone

No milestone

Development

No branches or pull requests

17 participants