**Backup and Restore of Kubernetes Applications using Heptio’s Velero**

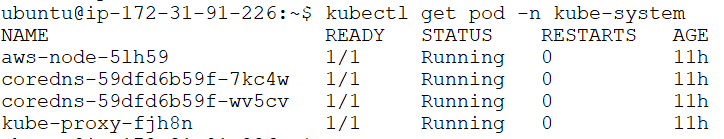
**Steps 1) => Prerequisite Just configure new Kubernetes EKS Cluster**

**1) Check node working fine**

# kubectl get node

**2) Check all IMP pod working related with network**

# kubectl get pod -o wide -n kube-system



**Steps 2) => Configure Velero with AWS S3 Bucket**

**1) Install AWS cli Package**

# sudo apt install awscli

**2) Configure Access Key and Secret key**

# aws configure

AWS Access Key ID [None]: AKANHV5A5EEQBOD

AWS Secret Access Key [None]: 3hQdUEN/6RxvzJN95qVL8148FbDh1aJdq

Default region name [None]: us-east-1

Default output format [None]:

**3) Check AWS S3 Bucket. It will prove that I have aws access.**

# aws s3 ls

**4) Create S3 bucket**

Velero requires an object storage bucket to store backups

BUCKET= velero-20

REGION= us-east-1

# aws s3api create-bucket \

--bucket velero-20 \

--region us-east-1

**Steps 3) => Set Permissions for Velero**

**Option 1: Set permissions with an IAM user**

**1) Create the IAM user:**

# aws iam create-user --user-name <User Name>

# aws iam create-user --user-name velero

**2) Attach policies to give velero the necessary permissions**

# cat > velero-policy.json <<EOF

{

"Version": "2012-10-17",

"Statement": [

{

"Effect": "Allow",

"Action": [

"ec2:DescribeVolumes",

"ec2:DescribeSnapshots",

"ec2:CreateTags",

"ec2:CreateVolume",

"ec2:CreateSnapshot",

"ec2:DeleteSnapshot"

],

"Resource": "\*"

},

{

"Effect": "Allow",

"Action": [

"s3:GetObject",

"s3:DeleteObject",

"s3:PutObject",

"s3:AbortMultipartUpload",

"s3:ListMultipartUploadParts"

],

"Resource": [

"arn:aws:s3:::velero-20/\*"

]

},

{

"Effect": "Allow",

"Action": [

"s3:ListBucket"

],

"Resource": [

"arn:aws:s3:::velero-20"

]

}

]

}

EOF

**3) Attach Policy to velero User**

# aws iam put-user-policy --user-name <User Name> --policy-name <Policy Name> --policy-document file://velero-policy.json

# aws iam put-user-policy --user-name velero --policy-name velero --policy-document file://velero-policy.json

**5) Create an access key for the user:**

# aws iam create-access-key --user-name <User Name>

# aws iam create-access-key --user-name velero

**6) Create a Velero-specific credentials file (credentials-velero) in your local directory: ( Velero User Name )**

# vim /home/ubuntu/credentials-velero

[default]

aws\_access\_key\_id=<AWS\_ACCESS\_KEY\_ID>

aws\_secret\_access\_key=<AWS\_SECRET\_ACCESS\_KEY>

**Steps 4) => Download & Install Velero Client**

**1) Download the latest official release's tarball for your client platform.**

# curl -LO https://github.com/heptio/velero/releases/download/v1.3.1/velero-v1.3.1-linux-amd64.tar.gz

# curl -LO https://github.com/vmware-tanzu/velero/releases/download/v1.3.2/velero-v1.3.2-linux-amd64.tar.gz

**2) Extract the tarball**

# tar zxf velero-v1.3.1-linux-amd64.tar.gz

# tar zxf velero-v1.3.2-linux-amd64.tar.gz

**3) Move the velero binary from the Velero directory to somewhere in your PATH**

# sudo mv velero-v1.3.1-linux-amd64/velero /usr/local/bin/

**4) Check velero home path**

# which velero

**5) If path not mention then export**

# export PATH=$PATH:/usr/local/bin/

**6) Check velero version**

# velero version

**Steps 5) => Install Velero Server**

If using IAM user and access key:

**1) Start the Velero server for Resource Backup Without snapshots**

# velero install \

--provider aws \

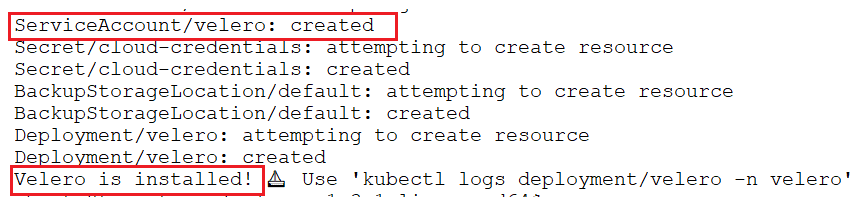
--plugins velero/velero-plugin-for-aws:v1.0.1 \

--bucket velero-20 \

--backup-location-config region=us-east-1 \

--use-volume-snapshots=false \

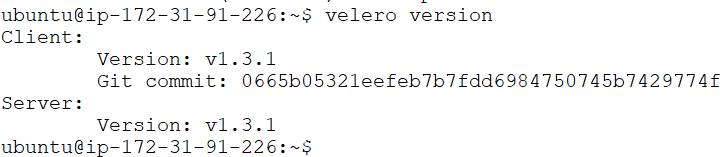
--secret-file ./credentials-velero



**2) Check velero version**

Note : if client and server both version will show mins velero configure properly

# velero version



=============================================================================

**Step 1) => Create Namespace and deployment for backup**

**1) Create Namespace**

# kubectl create ns it-department

**1) Create Nginx Pod in newly Created Namespace**

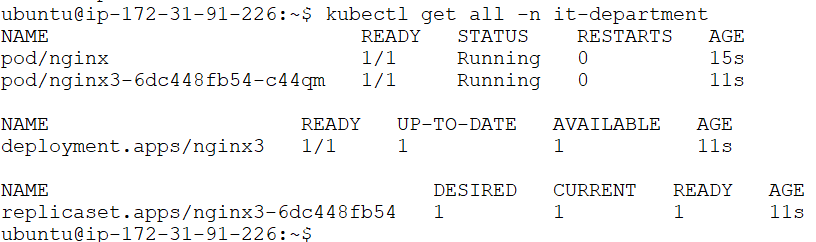
# kubectl run nginx --image=nginx -n it-department

**2) Create Deployment**

# kubectl create deployment --image=nginx nginx3 -n it-department

**3) Check all deployment of namespace**

# kubectl get all -n it-department

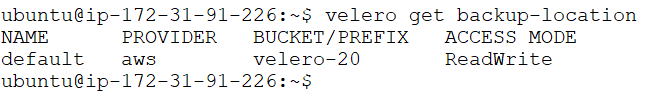


==========================================================================================

**Backup operation Velero**

**1) Get the velero backup location. ( AWS S3 Bucket velero-20 )**

# velero get backup-location

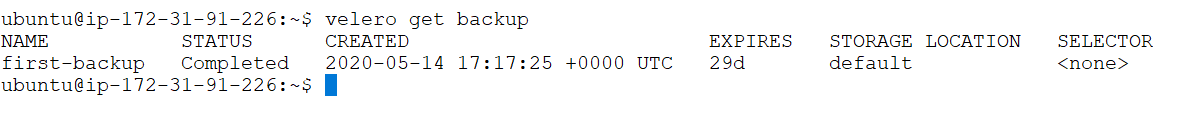


**2) Take backup of Entire Cluster**

# velero backup create firstbackup --include-namespaces it-department

**3) Check Backup Successfully created and stored to S3 bucket.**

# velero get backup



Note : I Can send Backup successfully to S3 Bucket   
Bucket Neme : velero20

**Manually Check webhook is working or not**

# curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World!"}' YOUR\_WEBHOOK\_URL\_HERE

# curl -X POST -H 'Content-type: application/json' --data '{"text":"Hello, World! yogesh"}' https://hooks.slack.com/services/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

**Velero Backup / Restore Notification**

**Step 1)=> Install Helm 2**

# curl https://raw.githubusercontent.com/kubernetes/helm/master/scripts/get> install.helm.sh

# ll

# chmod +x install.helm.sh

# ./install.helm.sh

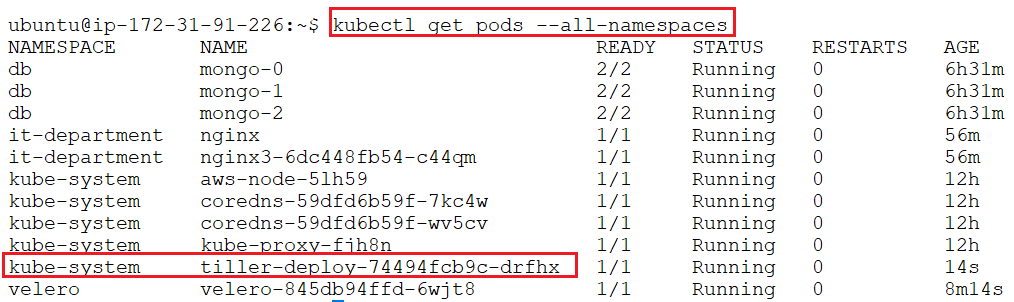
# ll

# kubectl -n kube-system create serviceaccount tiller

# kubectl create clusterrolebinding tiller --clusterrole cluster-admin --serviceaccount=kube-system:tiller

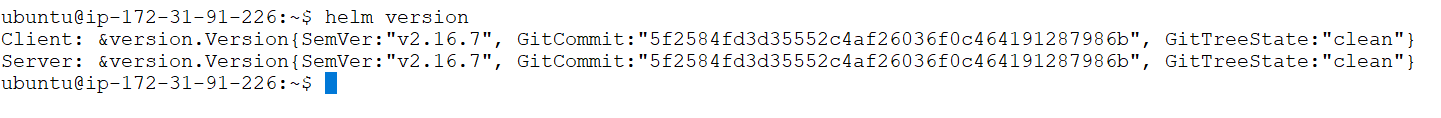
# helm init --service-account tiller

# kubectl get pods --all-namespaces



# helm status

# helm version



**Step 2) => Configure Velero Notification**

**1) Clone velero-backup-notification repo.**

# git clone <https://github.com/vitobotta/velero-backup-notification.git>

# cd velero-backup-notification/

**2) Install with helm**

helm install ./helm \

--name velero-backup-notification \

--namespace velero \

--set velero\_namespace=velero \

--set slack.enabled=true \

--set slack.webhook=https://hooks.slack.com/services/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 \

--set slack.channel=velero01 \

--set slack.username=Velero \

--set email.enabled=true \

--set email.smtp.host=smtp.gmail.com \

--set email.smtp.port=587 \

--set email.smtp.username=yogesh.bhagwatkar@nepallink.net \

--set email.smtp.password=Yogesh@1945 \

--set email.from\_address=yogesh.bhagwatakar@nepallink.net \

--set email.to\_address=yogesh.bhagwatkar@nepallink.net \

--set email.subject\_prefix="[Velero]"

**As per below output installation fail.**



Note : this installation not working. Deployment get failed

==========================================================================================

**Second Way to install Velero Notification**

**1) Delete all Backup from velero and S3 bucket.**

# velero delete backup first-backup

**2) Delete namespace of Velero. It will delete all setup of velero.**

# kubectl delete ns velero

**3) Delete velero installed by helm**

# helm del --purge velero-backup-notification

**4) Clone velero-backup-notification repo.**

# git clone <https://github.com/vitobotta/velero-backup-notification.git>

# cd velero-backup-notification/

**5) Create Velero Namespace**

# kubectl create ns velero

**6) Install with helm**

# helm install ./helm \

--name velero-backup-notification \

--namespace velero \

--set velero\_namespace=velero \

--set slack.enabled=true \

--set slack.webhook=https://hooks.slack.com/services/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 \

--set slack.channel=velero01 \

--set slack.username=Velero \

--set email.enabled=true \

--set email.smtp.host=smtp.gmail.com \

--set email.smtp.port=587 \

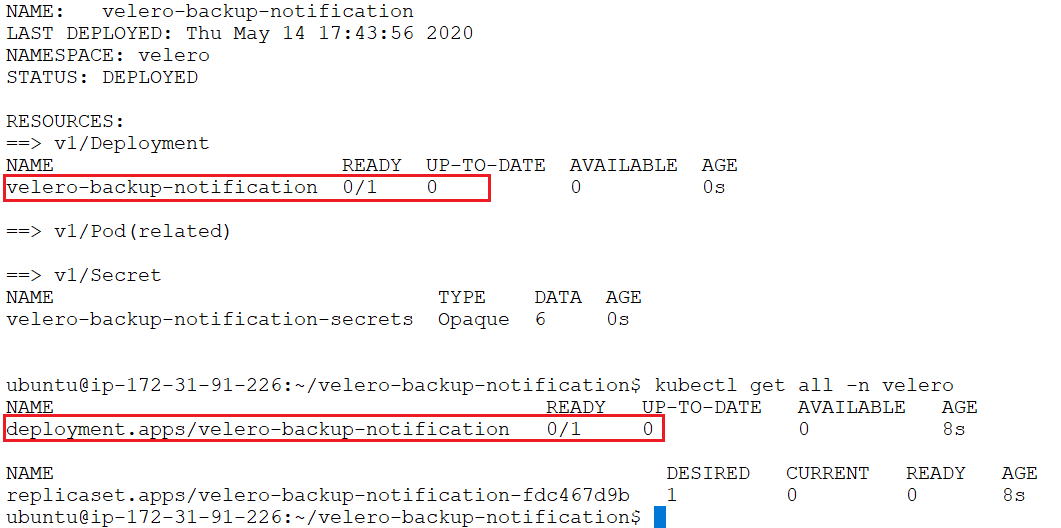
--set email.smtp.username=yogesh.bhagwatkar@nepallink.net \

--set email.smtp.password=Yogesh@1945 \

--set email.from\_address=yogesh.bhagwatakar@nepallink.net \

--set email.to\_address=yogesh.bhagwatkar@nepallink.net \

--set email.subject\_prefix="[Velero]"



Note : again velero notification setup failed.