

# Daily Business mit OpenShift

## Cluster User Administration

### Cluster-Admins verwalten

Um einem Benutzer cluster-admin Rechte auf der Plattform zu vergeben, müssen folgende Schritte beachtet werden:

1. Einloggen mit einem Benutzer, welcher bereits cluster-admin Rechte besitzt
2. Auflisten der berechtigten Benutzer der Rolle cluster-admin

```
# oc get clusterrolebinding | grep cluster-admin
```

```
...
cluster-admin          /cluster-admin          [cluster-admin]
system:masters
...
```

3. Benutzer berechtigen

```
# oc adm policy add-cluster-role-to-group cluster-admin [group]
```

- Benutzer wieder entfernen

```
# oc adm policy remove-cluster-role-to-group cluster-admin [group]
```

### Benutzer zu einem Projekt hinzufügen

Es gibt zwei verschiedene Arten, um einen Benutzer an einem Projekt zu berechtigen. Die drei meist verwendeten Berechtigungsvarianten sind folgende: admin, edit, view

Die erste Art wäre um GUI beim entsprechenden Projekt auf "View Membership" zu gehen. Darin können Benutzer mit Ihrem Suffix und der entsprechenden Berechtigung berechtigt werden. Development Platform > DailyBusiness > image2018-1-24\_13-18-36.png

Die zweite Art wäre via Command Line den Benutzer auf ein Projekt zu berechtigen:

```
# oc policy -n [Projekt] add-role-to-user [Rolle] [Benutzer]
```

### Projekt Berechtigungen anzeigen

Die Berechtigungen können entweder über das GUI unter "View Membership" angezeigt werden oder in der Kommandozeile:

```
# oc get rolebindings -n integration-service
```

Link zur Dokumentation:

[https://docs.openshift.com/container-platform/3.3/admin\\_solutions/user\\_role\\_mgmt.html#adding-a-role-to-a-user](https://docs.openshift.com/container-platform/3.3/admin_solutions/user_role_mgmt.html#adding-a-role-to-a-user)

## Cluster Project Administration

### Set Timeout on OpenShift route

Jenkins has some pages, like the system configuration or the plugin overview, that take up to a minute for displaying. With the default settings on the OpenShift reverse proxy the request will be canceled after several seconds. You may set a new timeout on the route for your Jenkins instance to wait for the page to load:

```
# oc annotate route <instance-name>-cing-helm --overwrite  
haproxy.router.openshift.io/timeout=60s
```

If a minute is not enough to load the page feel free to set the value higher.

## Cluster Administration

### Cluster Debugging

#### Kube Config wiederherstellen

Funktioniert der `oc get` command nicht mehr korrekt, oder besser gesagt wird hier ein Passwort verlangt, so ist die Kube Config verschossen.. Ein gültiges Backup befindet sich hier:  
`/etc/origin/master/admin.kubeconfig`

```
# cp /etc/origin/master/admin.kubeconfig .kube/config
```

#### Kube Config wiederherstellen

Funktioniert der `oc get` command nicht mehr korrekt, oder besser gesagt wird hier ein Passwort verlangt, so ist die Kube Config verschossen.. Ein gültiges Backup befindet sich hier:  
`/etc/origin/master/admin.kubeconfig`

```
# cp /etc/origin/master/admin.kubeconfig .kube/config
```

## Firmenspezifische Doku's

## Ressourcen von einem Projekt wiederherstellen

Bei der Post läuft täglich um 24:00 ein cronjob, der alle wichtigen Projekt-Ressourcen täglich exportiert:

```
#!/bin/bash
BACKUP_DIR="/var/openshift_backup/"
BACKUP_DIR_WITH_DATE=${BACKUP_DIR}/backup_$(date +%Y%m%d%H%M)

### Remove folders older than 30days
find ${BACKUP_DIR} -maxdepth 1 -ctime +30 -type d -exec rm -rf {} \;

### Setup
mkdir -p $BACKUP_DIR_WITH_DATE

### Project Backup
# Check if executed as OSE system:admin
if [[ "$(oc whoami)" != "system:admin" ]]; then
    echo -n "Trying to log in as system:admin... "
    oc login -u system:admin > /dev/null && echo "done."
fi

# Backup all resources of every project
for project in $(oc get projects --no-headers | awk '{print $1}')
do
    echo -n "Backing up project $project... "
    mkdir -p ${BACKUP_DIR_WITH_DATE}/projects/${project}
    oc export all -o yaml -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/project.yaml 2>/dev/null
    oc get rolebindings -o yaml -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/rolebindings.yaml 2>/dev/null
    oc get serviceaccount -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/serviceaccount.yaml 2>/dev/null
    oc get configmap -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/configmap.yaml 2>/dev/null
    oc get daemonset -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/daemonset.yaml 2>/dev/null
    oc get secret -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/secret.yaml 2>/dev/null
    oc get pvc -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/pvc.yaml 2>/dev/null
    oc get statefulset -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/statefulset.yaml 2>/dev/null
    oc get buildconfigs -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/buildconfigs.yaml 2>/dev/null
    oc get builds -o yaml --export=true -n ${project} >
    ${BACKUP_DIR_WITH_DATE}/projects/${project}/builds.yaml 2>/dev/null
    oc get imagestreams -o yaml --export=true -n ${project} >
```

```
${BACKUP_DIR_WITH_DATE}/projects/${project}/imagestreams.yaml 2>/dev/null
oc get jobs -o yaml --export=true -n ${project} >
${BACKUP_DIR_WITH_DATE}/projects/${project}/jobs.yaml 2>/dev/null
oc get route -o yaml --export=true -n ${project} >
${BACKUP_DIR_WITH_DATE}/projects/${project}/route.yaml 2>/dev/null
oc get services -o yaml --export=true -n ${project} >
${BACKUP_DIR_WITH_DATE}/projects/${project}/services.yaml 2>/dev/null
echo "done."
done
```

Dieser legt die Files in folgender Struktur auf dem ersten Master ab:

```
/var/openshift_backup/
├── backup_201806190000
│   └── projects
│       ├── appuio-infra
│       ├── default
│       ├── glusterfs
│       ├── kube-public
│       ├── kube-service-catalog
│       ├── kube-system
│       ├── logging
│       ├── management-infra
│       ├── ocp-grafana
│       ├── openshift
│       ├── openshift-ansible-service-broker
│       ├── openshift-infra
│       ├── openshift-metrics
│       ├── openshift-node
│       ├── openshift-template-service-broker
│       ├── openshift-web-console
│       └── patricks-testprojekt
├── backup_201806200000
│   └── projects
│       ├── appuio-infra
│       ├── default
│       ├── glusterfs
│       ├── kube-public
│       ├── kube-service-catalog
│       ├── kube-system
│       └── logging
└── .....
```

Restore erfolgt durch ein simples erstellen der Ressourcen. Zum Beispiel:

```
# oc create -f "Backup file in yaml" -n "namespace"
```

## OpenShift Gluster Volume extend - over heketi

Eine zusätzliche Terabyte Disk auf dem Gluster Cluster via Heketi einbinden: (siehe auch : <https://access.redhat.com/solutions/3164841>)

Login auf dem OpenShift GUI, Project "glusterfs", Pod "heketi-storage".

Zuerst die Admin Credentials über das YAML-File vom Pod "heketi-storage" auslesen.

Development Platform > DailyBusiness > image2019-1-25\_14-29-55.png

Auf der Heketi-Pod Konsole node und Cluster ID auslesen.

```
sh-4.2# heketi-cli node list --user admin --secret
p/J+0hZgDjwuMy/vZBoD74hP4CWmjWWE32ye6czr2aU=
Id:0680dabe91ee5a7f36da8cb6fe49cdd4 Cluster:7234de4476a10cb0d138e3fd3d387c40
Id:648f2115e99bf41fb78271acd55bd8f9 Cluster:7234de4476a10cb0d138e3fd3d387c40
Id:b016e52ee7378debc04427385f81cd82 Cluster:7234de4476a10cb0d138e3fd3d387c40
```

Jetzt kann das neue Device nach dem Schema "heketi-cli device add --name /dev/DISK --node NODE\_ID" für jede Node-ID des Clusters auf der Pod-Konsole eingegeben werden.

```
sh-4.2# heketi-cli device add --name /dev/sdc --node
0680dabe91ee5a7f36da8cb6fe49cdd4 --user admin --secret
p/J+0hZgDjwuMy/vZBoD74hP4CWmjWWE32ye6czr2aU=
Device added successfully
```

```
sh-4.2# heketi-cli device add --name /dev/sdc --node
648f2115e99bf41fb78271acd55bd8f9 --user admin --secret
p/J+0hZgDjwuMy/vZBoD74hP4CWmjWWE32ye6czr2aU=
Device added successfully
```

```
sh-4.2# heketi-cli device add --name /dev/sdc --node
b016e52ee7378debc04427385f81cd82 --user admin --secret
p/J+0hZgDjwuMy/vZBoD74hP4CWmjWWE32ye6czr2aU=
Device added successfully
```

Als Resultat der Erweiterung, sieht man nun auf dem entsprechenden Node mit dem Kommando "pvscan" die von heketi neu bereitgestellte Volumegroup.

```
[root@vosgel ~]# pvscan
PV /dev/sdc VG vg_5e4ce3cca0b61f43f749d3cef0447e2e lvm2 [999.87 GiB /
<979.72 GiB free]
PV /dev/sdb VG vg_bc2782bf157d2cde474ff55ae298715f lvm2 [999.87 GiB /
5.40 GiB free]
```

```
PV /dev/sda2    VG vgsys                                lvm2 [<79.47 GiB /
0    free]
Total: 3 [2.03 TiB] / in use: 3 [2.03 TiB] / in no VG: 0 [0    ]
```

Last update: **2020/12/10 14:07**