* cd /var/log/httpd
* archive the logs. Send it SMW out of the system (S3 Backet). Clean the logs
* create s3 bucket
* archive logs. Move to temp directory. Clean /var/log/httpd
  + tar czvf web01-logs.tar.gz \*
  + cat /dev/null var/log/httpd/access\_log
* Install awscli.
* create IAM user (access/secret keys, S3adminaccess)
* aws configure
* send logs-archive files to S3 backet
  + aws s3 sync /tmp/logs/ s3://web-logs-backet777
* clean local log archives/logs
* S3 Bucket instead of telegram share. Private keys. Plans. Python sync script.
* S3 Backet - Access
  + Role -> based EC2 instances
  + IAM access keys -> based users
  + Set access policies on S3 backet -> for LBs
* Make Github via password. Make Github private
* stream logs to dashboards (live logs)
* Remove aws credentials: rm -rf .aws/credentials & now test access to S3
* create roles access (instead of IAM). S3fullacess.Cloudwatchlogfullaccess.
* Attach role to EC2 & test access to S3. Aws s3 ls
* Install cloudwatch logs agent on EC2
  + <https://docs.aws.amazon.com/AmazonCloudWatch/latest/logs/QuickStartEC2Instance.html>
* Add our logs. nano /etc/awslogs/awslogs.conf
  + [web\_access\_logs]
  + file = /var/log/httpd/access\_log
  + log\_stream\_name = access\_logs
  + log\_group\_name = web\_access\_logs
* Restart awslog service
* Create metrics – Hacker\_IP
* Generate logs for LB and Save it
* Create LB – Classic. Create Access policy on S3. Configure access logs on LB. Save logs in S3 Bucket.
* Create Access policy on S3
  + <https://docs.aws.amazon.com/elasticloadbalancing/latest/classic/enable-access-logs.html>
* The S3 bucket must be located in the same region as the load balancer.
* <https://docs.aws.amazon.com/vpc/latest/userguide/vpc-network-acls.html>