ITT3 SS special subjects

SS3 servers

UCL University college

Contact:
Morten Nielsen <mon@eal.dk>

2018–11–28

Contents

1   Project outline

This special subject is about automation, virtual machines and automatic handling of testing, building and other administrative tasks.

1.1   Network overview

For this special subject, we will build a network, which is not simple, but well within reach of ITT3.
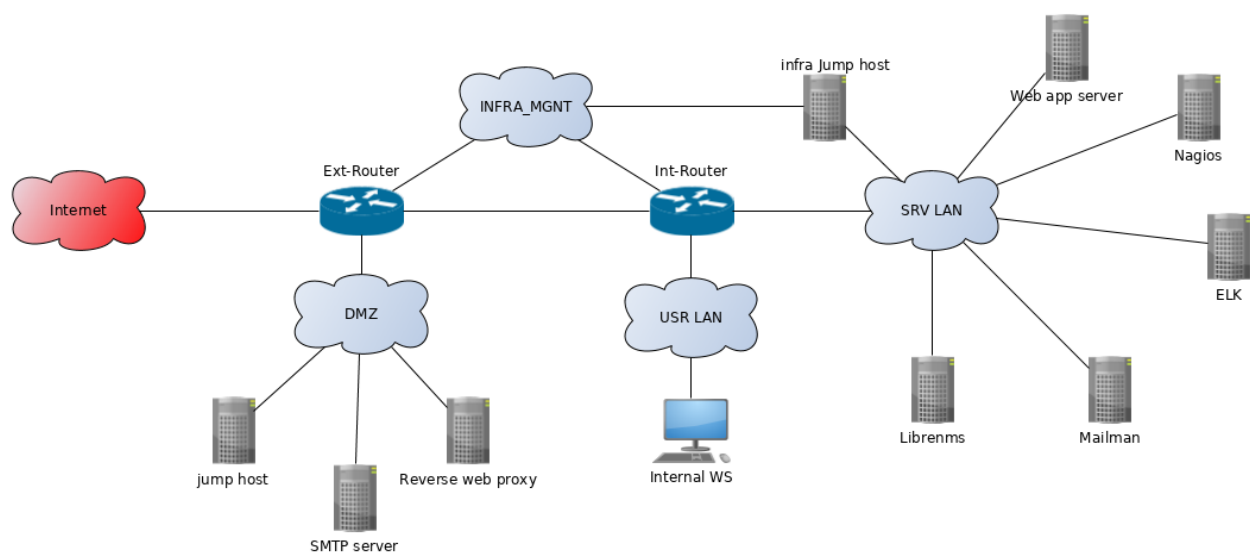
We have 4 subnets

- `DMZ` - for servers which may be accessed directly from the Internet
- `USR LAN` - for user workstations
- `SRV LAN` - for intenal servers
- `INFRA_MGNT` - non-routable subnet for configuring the infastructure devices, ie. the two routers

We have lots of servers

- `Jump host` - the host that is used to acces the network from the internet
- `SMTP server` - for incoming and outgoing smtp traffic
- `reverse web proxy` - a proxy to give access to internal webservers
- `internal ws` - some internal workstation. We will include one virtual workstation, and might include an option for hooking in physical workstations also
- `infra jump host` - jump host for accessing the infrastructure management network
- `web app server` - the sererrunning our webapp

- `nagios`, `librenms` and `ELK` - monitoring and log servers
- `mailman` - the mailing list management server

The webapp in question will pull data from the mailman server and present it in a nice way to users.

1.2   Other notes

We will divide whte world into production, testing and development. Development will mainly be on people laptops.

We will use gitlab and their tools for automating tasks in this environment.

We will automate creating and testing of the VMs and build VMs that may be used by others.

1.3   Outline of the 5 weeks

Weeks as currently planned are as follows.

- Week 45: SS3 startup, DevOps
- Week 46: Fontys week
- Week 47: Reproducability
- Week 48: Merging
- Week 49: Autodeploy
- Week 50-51: Exams

2   Week 45 DevOps

2.1   Goals of the week(s)

Pratical and learning goals for the period is as follows

2.1.1   Practical goals

- None at this time

2.1.2   Learning goals

- DevOps
- Level 1: Know what DevOps and CI is
- Level 2: Able to script installation of a server
- Level 3: Able to automatically generate VM images

2.2   Deliverable

- Weekly presentations Friday at 12:15
- Probably some personal document update
- Peergrade reviews

2.3   Hands-on time

- Exercise 1 (LG lvl 1)

Using the Feynman technique[1], write 250 words about DevOps and CI. Self-evaluate afterwards by highlighting formulations and concept where you feel weak.

You will review each other using peer grade.

- Exercise 2 (LG lvl 2)

Install a server. The specific server is different from each group, and we decide on class who builds what.

Make a script, runable by root that installs the server from a well defined starting point. Remember to include a `readme.md` file to describe how it is used.

This must be put on gitlab, and accessible. Make sure that this works as run by root on the specified starting point.

    git clone $REPOLOCATION
    cd $REPONAME
    sh ./runme.sh

We will automate this, so you must adhere strictly to the format.

---

[1] https://www.youtube.com/watch?v=tkm0TNFzIeg

- Exercise 3

Use `packer` to create a VM based on your script.

2.4   Comments

- Thursday, we discuss peergrade from last week
- See the project introduction for the list of servers.
- Friday, Jonas from UFST will show up and tell us about how they work with DevOps.

3   Week 46 Fontys week

We have foreign visitors this week, and you will be project managing a group.

See the Fontys document for details.

4   Week 47 Reproducability and testing

4.1   Goals of the week(s)

Pratical and learning goals for the period is as follows

4.1.1   Practical goals

- You have access and have tested to build procedures of others

4.1.2   Learning goals

- Reproducability and testing
- Level 1: Know what building and testing means in CI
- Level 2: Able to implement a semi-automated workflow for virtual machines
- Level 3: Able to implement fully automated build and test environment

4.2   Deliverable

- Weekly presentations Friday at 12:15
- Probably some personal document update
- Peergrade reviews

4.3   Hands-on time

- Exercise 0

We will do a pre-mortem meeting. (3 groups)

1. (20 min) Image that we fail in implementing the project. In the group, come up with as many reasons as possible for that to have happened.

2. Select the top 5 most relevant.

   - `relevant` might be equivalent to `most likely` in this context
   - exclude stuff we don't have any control over, e.g. meteor showers.

3. Write them on the whiteboard

4. (30 min) On class, we will discuss how to prevent the problems from arizing.

   - this includes who does what.

- Exercise 1 (LG lvl 1)

Using the Feynman technique[2], write 250 words about how to implement the CI concepts and methodology, using this class and project as an example. Self-evaluate afterwards by highlighting formulations and concept where you feel weak.

You will review each other using peer grade.

- Exercise 2 (LG lvl 2)

Implement a workflow similar to the one you described in exercise 1 on your specific server.

In this iteration, there will probably be manual steps, perhaps for building, perhaps in the test phase.

---

[2]https://www.youtube.com/watch?v=tkm0TNFzIeg

- Exercise 3

Automate building and testing of the server using gitlab ci pipelines.

Start with exercise 3 from last week, if applicable.

4.4   Comments

- Read about CI here[3] and here[4].
- Read about pre-mortem here[5] and here[6].

5   Week 48 Merging to testing

5.1   Goals of the week(s)

Pratical and learning goals for the period is as follows

5.1.1   Practical goals

- We have set up a system on a test blade server.

5.1.2   Learning goals

- Merging to testing
- Level 1: Know about automated building and the steps needed
- Level 2: Able to build a virtual machine and put it into the test environment
- Level 3: Able to implement automated build

5.2   Deliverable

- Weekly presentations Friday at 12:15
- Probably some personal document update
- Peergrade reviews

---

[3]https://codeship.com/continuous-integration-essentials
[4]https://www.tutorialspoint.com/continuous_integration/
[5]https://www.riskology.co/pre-mortem-technique/
[6]https://www.mybeeye.com/blog/pre-mortem-effective-tool-to-prevent-failure

5.3   Hands-on time

- Exercise 1 (LG lvl 1)

Using the Feynman technique[7], write 250 words about packer and how it fit with CI, using this class and project as an example. Self-evaluate afterwards by highlighting formulations and concept where you feel weak.

You will review each other using peer grade.

- Exercise 2 (LG lvl 2)

Use packer to build your server from the base image you defined.

- Exercise 3

Automate building and manual deployment. This will require discussion on class on how we do the actual deployment and the actual building.

5.4   Comments

- Read about packer here here[8], using esxi to install a VM is here[9] and using an existing image is here[10]
- To add stuff to an ESXi it seems that we should use OVF. Pakcer supports that (see here[11]) and something with ovftools (see here[12])

6   Week 49 Autodeploy

Details to come

7   Weeks 50 and 51 exams

We have exams these weeks.

---

[7] https://www.youtube.com/watch?v=tkm0TNFzIeg
[8] https://www.packer.io/intro/
[9] https://www.packer.io/docs/builders/vmware-iso.html
[10] https://www.packer.io/docs/builders/vmware-vmx.html
[11] https://www.packer.io/docs/builders/virtualbox-ovf.html
[12] https://www.vmware.com/support/developer/ovf/

8   Mandatory elements

To help keep you on track, and to maintain an apropriate level of teacher contact, there are some mandatory elements.

- All weeks will include presentations on class.

  This is a presentation with the class as *audience* and you must have decided in advance the *purpose* of the presentation

- Project management

  You will create a gitlab project for project management and for configurations.

  Remember to invite the teachers.

- All weeks will include a meeting with the teacher(s)

  This is a 10-15 minute meeting, where you present status on tasks and milestones, and if you have specific questions

9   End-of-SS report

Each SS ends with a report hand-in. The report must document, that the student is able to:

- Document a project or process
- Learn new skills and/or gain new knowledge
- Explain technical or other relevant material within a given context
- Present technical or other relevant material within a given context
- Do structured project management

The report will includes links and references to relevant material, either online links or appendices. The former is prefered.

Since Gitlab is used for SS, links to gitlab resources like design documents, tests and more may be relevant.