COLLECTIVE INFRASTRUCTURES PANEL

PRACTICING PROTOCOLS WORKSHOP

Please login to this wifi network:

PW:

INTRO IN-GRID

In-grid is a trans*feminist collective of artists/educators/technologists working in and around digital infrastructures. In-grid has been working together since 2019, and is made up of a fluid group of people dispersed globally.

COMBINED PANEL

This combined panel takes a critical look at digital "cloud" infrastructures. It explores how collectives and communities counter these centralised models through their situated relations, methods and technical practices.

The **Combined Panel** takes the form of:

- A Panel
- A workshop <-- We are here
- A Hosted Blog <-- We will be posting to here

PANEL

For the panel we brought together a range of researchers and practitioners. Each panellist shared their nuanced context and how their collectives*communities are making transformations in infrastructures. We did this to see how system admins and community organisers are forming a diversity of nuanced and situated protocols and ways of collaborating otherwise.

WORKSHOP OUTLINE <-- (WE ARE HERE)

Today's workshop aims to bring together the combined panel's conceptual elements with material socio-technical practice. In this session we will enact the creation of a speculative set of protocols that aim to question the relational complexities of editing, updating and maintaining a collective server, and learning some basic technical skills along the way!

The end goal is to have collectively edited set of protocols that we publish to an html page on the server.

BLOG <-- (WE WILL BE POSTING TO HERE)

The blog element aims to extend conversation from the panel through our own federated infrastructure, subverting and hacking the hierarchies of this conference. It brings this research together in a space that can sit both inside/outside*beyond the conference.

Access it at: ci.servpub.net

PROTOCOLS

We will also be making and editing a set of protocols as we work through the steps of this workshop.

The protocols will be hosted on an html file on the server, and is accessible at this URL page: https://servpub.net/ci_protocols.html

Until then we will edit them on this pad: https://pad.vvvvvvaria.org/workshopProtocols

WHAT ARE THESE PROTOCOLS?

When it comes to making collaborative community driven sociotechnical infrastructure, there is always a complex network of relations that we have to work through. Negotiating every relation when we enter a space could be very complex and time consuming, as we will see, but setting protocols helps us firm up ways of working and relational conventions that ease this flow. Through setting these protocols together we mean to disobey the norms of network infrastructures and actively re-scope how we come to these systems and relations together.

DIAGRAM: OF SOCIO-TECHNICAL RELATIONS OF SERVPUB PROJECT

servpubDiagram2024

1. INTRO TERMINAL SOFTWARE

Lets give a little context of working with computers

GUI

So most people are probably used to working through a **graphical user interface** (GUI), which is a relatively recent way of interfacing with a computer. This is great for accessibility but also tends to scope and direct the abilities and capacities you have on your computer, as in there are only so many buttons or things you can show in a communicable way.

TERMINAL

howdy.png

Terminal is a more classic way of working with a machine, and roughly speaking is the step between punch-card/magnetic tape style computing and a GUI. It is a very simple but powerful interface that lets you run basically any command on your system through a text interface. Compared to a GUI this space is more "risky" as you can do anything from anywhere, such as deleting the entire hard disc (not advised). It also means that there are no limitations to your system administration so you have access to a granular level of working.

PROTOCOLS

When entering into this more "risky" space on a computer network, it is good to set some intentions and work out how to move through this space with care. As we go though these next steps getting into terminal have a think about what it means to be in this place of power/risk, and what relations you think you need to care for and form protocols for?

It can also be a great place to think about if/how your system let's you into this place of power, and what relations this empowering brings with it.

2. ACCESSING TERMINAL

This depends on your platform, so...

MAC & LINUX

On a Mac it is already installed! So all you need to do is:

1. Search up Terminal
2. Open it!

WINDOWS

There are similar programs to terminal already installed, but not the same language (bash), so the easiest way to get going is:

1. Install git Bash from HERE
2. Then open it up!

3. INTRO TO TERMINAL COMMANDS

As we said before terminal is a text/syntax-based way of interfacing your machine and we are going to introduce a few basic commands and concepts to get us going today.

SAY HI!

Let's simply use `echo` to get our terminal to say HI!.

To do this use:

```
echo <message>
```

e.g.

```
echo HI!
```

WHO ARE YOU?

Now lets see who your computer!

Show the name of the currently logged in user:

```
whoami
```

Show the name of the machine is given on the network:

```
hostname
```

STRUCTURE + NAVIGATION

Much of working on the command line involves navigating around your system without the use of a Graphical User Interface (GUI).

However, we will essentially be navigating around normal computer folder hierarchies like below:

```
root/
├─ directoryname/
       ├─ anotherdirectory/
                  ├─ aFileName.txt
```

Directory : also known as a folder

Print working directory that you are currently in:

```
pwd
```

List current directory contents:

```
ls
```

MOVING AROUND

Now to move between directories/folders we use `cd` like so

```
cd <directory_name>
```

e.g.

```
cd folder1
```

We can use a folder that was shown with `ls` in the last step! You can also check whats in the new folder with `ls` to see that you have moved.

WRITING TERMINAL PROTOCOLS

Prompt

In this unfamiliar space how do we keep track of where we are, what we've done & what commands we may need in the future?

...or any other concerns!

3. INTRO TO SSH

Secure Shell (SSH) is a way of working on another machine from your own, working in their shell (bash).

It basically lets you work on another computers terminal remotely. This is the way you will work on most servers or remote/faceless computers (computers without a screen).

SSH PROTOCOL

SSHDiagram

The SSH protocol enables us to talk securely to our server by encrypting and authenticating all of our communication/ interfacing through the network.

Traditionally tech has a metaphor of SSH as a secure hand shake protocol, but it will be great to think about what metaphor or relation we want when it comes to crossing boundaries and spaces.

Do we want to hand shake or do we think . . . would enable . . .

INSTALLING SSH

Again this may be a bit different on different systems, as some will have this already installed and some will need to install it.

if you are on **Mac** or **Linux** you may need to install it with:

```
sudo apt install openssh-client
```

If you are on gitbash on windows, it should be installed.

GETTING THE IP ADDRESS OF THE PI

To SSH into the pi we need to know what the IP address of the pi is.

An IP address is basically the postcode/address we will find the pi at on the network.

To do this on the pi we write:

```
hostname -I
```

This will print out something like `192.168.0.8`, and we need to save this to SSH into the pi

SSH INTO THE PI

Now that we have the IP address, any computer on that network can request to SSH into the Pi through these commands:

```
ssh <user_name>@<ip>
```

eg.

```
ssh ci@192.168.8.120
```

You are now "logging into" the pi, so it will ask for the password here and that is . . . . . .

ARE WE IN THERE?

You should now see your terminal username change from your laptop's local user to something like:

```
ci@servpub~$.
```

If you want to exit out of the pi (we don't atm) type:

```
exit
```

Hit enter. You will need to have exited the pi before setting up your ssh key.

SECURE SSH KEYS

We won't do it here as it takes a minute too long, but to do this securely we would set up a set of keys which, one private and one public, that will do the "hand shake" instead of a less secure password.

WRITING SSH PROTOCOLS

❓ **Prompt**

How do we navigate another machine through our own "interface"?

What are the relational agreements to being on this other machine?

4. INTRO TO SHARED SERVER

Caring for a server includes routine tasks like updating/ upgrading the system software, tiding up files, creating new users & giving access permissions, paying attention to errors and bug reports.

Sharing a server space presents the added complexity of doing all of the above with consideration other collectives' and/or groups' pre-existing protocols (eg. Systerserver)

ConnectionsDiagram

5. ENTERING INTO SUDO

Sudo enables us to take on a **superuser role**. This means that we have all the security permissions for that system. This is a "very risky" role, as here you can do anything on the server/ computer. With sudo you can actually delete the entire system or make other similar drastic commands, but it also means that we have the ability to manage and configure our system to our needs.

When we come to writing protocols for sudo, what does it mean for us to be a "superuser" or an admin, and what care and consideration should we take on in these protocols?

Usersdiagram

BECOMING SUDO

To become sudo you just need to:

```
sudo su
```

And then enter the password when prompted.

WRITING SUDO PROTOCOLS

Prompt

Super User can Do! a lot... what permissions are we comfortable navigating?

How intimate are we with our machines?

6. INTRO TO TMUX

Tmux is what is called a multiplexer, which is a software that lets us collectively work one terminal session.

This is how many collectives and collaborative networks work together on servers in one space. It lets us edit and run commands in one space and take responsibility and practice our infra becoming together.

When we come to writing protocols we might want to think about what it means to be one user together, and how can we keep everyone at the same pace and comfortable levels of understanding.

MAKING A COLLECTIVE SESSION

Now we are all `sudo` we can initiate a shared session to work in together.

To do this we use:

```
tmux new -s hugs
```

Now we need to name our collective space. If we think back*forward in this workshop, what are the intentions, dynamics and relations we want to collectively name this space with?

JOINING OUR SESSION

Now we have an active session we can join it by just typing:

```
tmux a -t hugs
```

Rember we have to be sudo to join this!

SAYING HI TO EACH OTHER!

Now we can say hi to one another and say cute messages in the space by:

```
echo <message>
```

Say something to welcome each other into this session, you may need to take turns.

COLLECTIVE WORKING

As we saw even each of us writing an echo command takes a little bit of choreographing. When this gets more complicated, like writing complex configuration files, you can imagine it gets even more tricky.

So here we need to think about how we might manage collective editing. What operational metaphor might we use to describe the relationship between the person typing and the others? Who and how do we choose to type? How do we take care of every voice in that conversation?

7. COLLECTIVELY EDITING

We've created a simple html document to edit. We need to navigate to that file, open and edit it.

First lets navigate to the folder its inside and list its contents.

```
cd /var/www/pubdoc
ls
```

We should see a file called:

`ci_protocols.html`

EDITING THE HTML

Now we are in the right folder we can edit the file with **nano** which is a terminal base text editor.

To open the file with **nano** we do:

```
nano ci_protocols.html
```

This will open the file in the terminal.

THIS SHOULD LOOK SOMETHING LIKE THIS

nanoOfHTML

Tip

Being terminal you have to navigate with the keyboard, and also copy and past by right clicking (not "CNTRL+C/P")

ADDING TO THE FILE

From here we can then add text and images using basic HTML syntax.

So for now add a message or paste in some ascii art with the help of these links:

ascii text gen

ascii image gen

SEEING OUR PAGE!

Now if we return to the file online at:

https://servpub.net/ci_protocols.html

we will see our beautiful inputs.

DO WE WANT TO EDIT OR CHANGE THE COLLECTIVE EDITING PROTOCOLS?

8. EXIT SAFELY

When we finish up working together there are lots of things we can do to make it easier to return to our work, make it legible to others, as well as caring for the infrastructure we're working in. This could include careful documentation, understanding how to save and record edits, and exiting programs in their nuanced ways.

EXITING TMUX

We've finished working together and now we would like to leave our TMUX session. This is a little more complex than it might appear. Leaving a session using an 'exit' command will delete the session for others, and you will lose the record of your edits together. Instead we can either:

Close the entire terminal session window, or use Ctrl-b. This will retain the terminal session for you to use again in the future.

EXITING SAFELY

Let's consider collectively how to ensure we exit safely, using a door rather than an ejector seat. How does this action differ from exiting other softwares and how does this attention affect our relationships.

WRITING THE PROTOCOLS

prompt

How do we negotiate the space of working collectively as one user?

What other considerations need to be voiced?

9. PUBLISH PROTOCOLS

Now we have all the protocols ready lets add them to the HTML file with **nano** again.

Once we have edited the HTML, we can launch our page in a browser!

At https://servpub.net/ci_protocols.html

WE DID IT!

Practising infrastructuring collectively can be (intentionally) a slow, careful and unscalable process, so it's important to acknowledge our incremental changes, achievements and presence.

REFLECTIONS

RESOURCES FOR MAINTAINABLE PRACTICES

Our (ongoing) documentation https://github.com/In-grid-collective/servPubWiki.git